ANGLIA RUSKIN UNIVERSITY

FACULTY OF SCIENCE AND ENGINEERING

A NEW SECURITY SCHEME AND LIGHTWEIGHT ENCRYPTION ALGORITHM FOR VOICE OVER WIRELESS NETWORKS CONNECTIVITY TO INTERNET

FIRAS IBRAHIM HAZZAA

A Thesis in partial fulfillment of the requirements of

Anglia Ruskin University for the degree of a

Doctor of Philosophy

Submitted: May 2019

Acknowledgments

I am eternally grateful for the support that numerous people who provided me with exceptional support, encouragement and wisdom who have helped me along the way.

First and foremost, I thank my God "*Allah"* for this achievement. And I am grateful to my first supervisor Dr.Sufian Yousef for his efforts in providing me invaluable suggestions and kindness and continuous support during this work. I am deeply grateful for my supervisory team Dr.Erika Sanchez and Dr.Nada Ali for their intelligent guidance and fervent support. I would also like to thank Prof.Marcian Cirstea the head of computing school at ARU and Dr.Antesar Shabut for their support in Global events and publications.

I am deeply grateful to the Ministry of the Higher Education and Scientific Research in Iraq for funding this grant during the whole research period, this work would not be possible without their support.

I am also thankful for the support of my family for their unconditional love, understanding and support, my parents, my wife and my kids Hiba and Yazen. They have been an endless source of great assistance, joy, and love to finish my Ph.D.

Sincere thanks to my friends, colleagues, and collaborators: Philp, Mohammed Al-Amin, Mohamed Fakheralldeen, Folayo, and many others. Finally, Many thanks for the National Institute of Standard and Technology and for all institutes and organizations that support during the research time.

Declaration

I declare that this thesis is my own original work, except for references. This thesis has not been submitted, in whole or in part, to any universities, except for the qualification of Doctor of Philosophy (PhD) at Anglia Ruskin University.

Firas I. Hazzaa

May 2019

ANGLIA RUSKIN UNIVERSITY

ABSTRACT

FACULTY OF SCIENCE AND ENGINEERING

DOCTOR OF PHILOSOPHY

A NEW SECURITY SCHEME AND LIGHTWEIGHT ENCRYPTION ALGORITHM FOR VOICE OVER WIRELESS NETWORKS CONNECTIVITY TO INTERNET

FIRAS HAZZAA

MAY 2019

The security system designed for Internet of Things IOT should be able to detect and prevent both internal and external attacks. It should be noted that not all connected devices have enough computational power. That means tasks like encrypting data are going to be impossible and any type of security must be lightweight. The privacy of the information on IOT needs a reliable security system that prevents unauthorized access to private data on the network. Cryptographic mechanisms must be smaller and faster but with no reduction in security level.

In this work, a new security system for voice over wireless networks is being developed and tested. New encryption algorithms have been developed to meet the Quality of Service QoS requirements of voice traffic and to be suitable for wireless devices. The goal of the research is to reduce the execution time and Energy consumption of the encryption process and at the same time at least maintain or increase its security level. The proposed scheme uses similar methods used in Advanced Encryption Standard algorithm AES, with some changes and enhancements considering the limitations of the wireless device. The new technique introduces triple key usage in this algorithm, making it difficult to break and is more secure. A range of simulation scenarios are setup; testing data is analyzed to test delay, energy, and security.

The test results show significant improvements in new design metrics. Also, the comparison between the new algorithms and the standard one shows a significant amount of time and energy consumption reduction being achieved (approximately 30% - 35%), with a high-level of complexity. The results provide and validate a framework for the implementation of security methods.

The proposed algorithm is more suitable for wireless devices with limited resources, and it achieves a considerable trade-off solution (balance) between security and QoS, thus it exhibits its applicability for any wireless networks where the resources are limited.

Key Words: lightweight cryptography, secure communication, voice encryption, AES encryption, wireless network security.

Contents

List of Figures

List of Tables

Abbreviations

AES	Advanced Encryption Standard
Crypto	Secret
CPA	Chosen Plaintext Attack
CCA	Chosen Ciphertext Attack
Cipher	Encrypted
CPU	Central Processing Unit
DES	Data Encryption Standard
Decipher	Decrypted
erfc	complementary error function
FTP	File Transfer Protocol
IOT	Internet of Things
IP	Internet Protocol
ICT	Information and Communication Technology
IEEE	Institute of Electric and Electronic Engineers
IPsec	Internet Protocol Security
LEA	Lightweight Encryption Algorithm
MANET	Mobile Ad Hoc Network
MixCol	Mixed Columns
MB	Mega Byte
NIST	National Institute of Standard and Technology
P2PSIP	Peer to Peer Session Initiation Protocol
QoS	Quality of Service
RC4	Rivest Cipher 4
S-box	Substitution box
SubByte	Substitution Byte
SSL	Secure Socket Layer
SIM	Subscriber identity module
SDD	Symmetric Dual key Dynamic block
TKE	Triple Key Encryption
VoIP	Voice over Internet Protocol
VM	Virtual Machine
VANET	Vehicle Ad Hoc Network
VPN	Virtual Private Network
WANET	Wireless Ad Hoc Network
WSN	Wireless Sensor Network
WMSN	Wireless Multimedia Sensor Network
WLAN	Wireless Local Area Network
WEP	Wireless Equipment Protocol
3DES	Triple Data Encryption Standard
.wav	Audio wave format
3G	Third Generation
4G	Fourth Generation
5G	Fifth Generation

Copyright

This work may:

- ✓ Be made available for consultation with Anglia Ruskin University Library or,
- ✓ Be lent to other libraries for the purpose of consultation or may be photocopied for such purposes,
- ✓ Be made available in Anglia Ruskin University's repository and made available on open access worldwide for non-commercial educational purposes, for an indefinite period.

1 Chapter One: Introduction

1.1 Introduction:

Security and Cryptography are the two aspects to protect communication and data. The security implements the protocols that govern what goes where, when and how, while cryptography provides the methods of applying such security. According to (Morgan, 2017) the cost of cybersecurity reached $84 bn in 2015, and it would expect to reach $170 bn in 2020. Therefore, most of the businesses are giving huge attention to security concerns.

The wireless network is a set of wireless nodes that connect with wireless links and are not connected by cables of any kind. The use of a wireless network enables enterprises to avoid the costly process of introducing cables into buildings between different equipment locations (Sharma , et al., 2017). However, the security of information over wireless networks is vulnerable to different kind of attacks, because of the nature of this network and any node can join or leave wirelessly. Cybercrime is increasingly growing and existing practical models to tackle cybercrime are ineffective in stopping the increase in cybercrime (Jahankhani, et al., 2014).

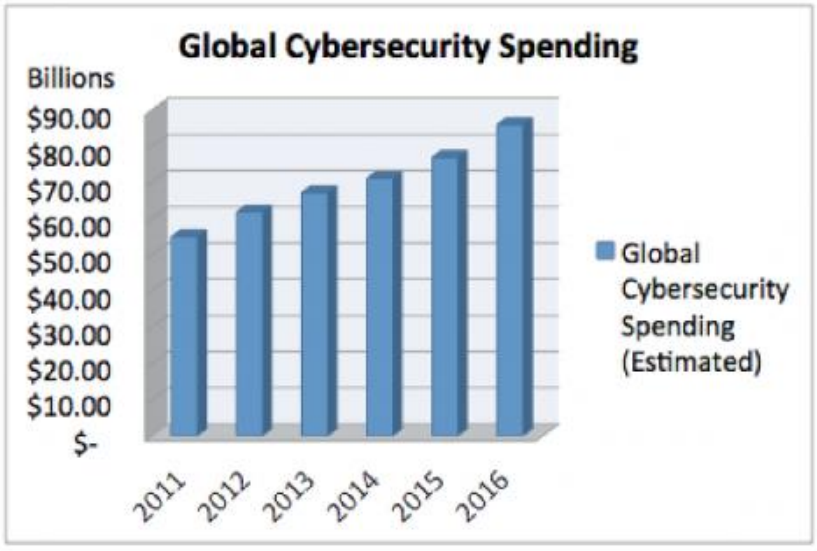


Fig. 1-1 Security Spending (Gartner, 2017)

The main factor of security is confidentiality, which can be achieved by encryption. Encryption algorithms have been used in many applications and protocols. They have been used to encrypt data transferred into the network and could be used to encrypt the IP address as well (Hazzaa, et al., 2018). The routing information and request-response packets between

wireless nodes could also be encrypted. These features have made the encryption essential for security. However current cryptography algorithms are not suited wireless environment because of nodes limitation such as time and Energy consumption. This research will suggest a new encryption scheme to be suited to these networks.

This thesis is different from pure cryptography thesis, which would focus on the mathematical computational operations. This thesis instead discusses wireless network security and how the cryptography operations related to their function. It also explains the network limitation and the cost of the implementation of current cryptography schemes.

In this thesis, the **aim** is to find an efficient security system for encrypting the voice and multimedia information data without expensive cost such as, delay, throughput and the power consumption. In addition to propose a lightweight encryption algorithm with the same level of security compared with the standard algorithm. For this purpose, many experiments will be required to determine QoS in a wireless network and to test the effect encryption has on QoS parameters. The impact of this work is significant because there is still a huge threat to the privacy of individuals and at the same time the cost is still high. Therefore, best trade-off solutions are highly required for the wireless environment. This study will reduce the cost of the security without affecting security level.

1.1.1 Brief Overview for Wireless Networks

Wireless networks are key enablers of next-generation communications. These networks can be designed and reconfigured dynamically and they can be mobile, standalone or internetworked with other networks (Thomas & Robertazzi, 2017). Wireless networks are very important in the IoT environment and are deployable. For instance, Mobile Ad-hoc Networks are established by a group of autonomous nodes that link with each other by creating a multi-hop radio network and maintain connectivity in an infrastructure less manner. They allow data, voice and video communications over a wireless channel (Panwara, et al., 2016). The key role of WMANET is to facilitate wireless and mobile communication service without having a previously set up infrastructure and without using the expensive service provider network, it makes the service available anywhere, any time.

Voice and multimedia are very important in wireless networks and play crucial role in allowing people to communicate, like a battlefield and emergency operations, so it’s important to secure this traffic in such networks.

1.1.2 Brief Overview for the Security and Cryptography

Security of the connections between devices and networks is crucial. The important challenges for supporting multimedia applications in the Ad hoc network are the security issues (Hazzaa, et al., 2018). Unfortunately, conventional multimedia traffic management algorithms, developed mainly to guarantee less delay while usually neglecting security requirements. According to (Stallings, 2017; Zhou, 2010; Zhao, et al., Jan. 2009) applications and users can be a source of security threats. (Liang & Chao, 2011) claim that it is very important to adopt a security strategy to protect critical voice and multimedia applications. Many previous researchers (Nagendra & Sekhar, 2014; Ali, et al., 2014; Binod & Hyuk, 2010) adopted security algorithms that were weak enough to deal with the constraints of the quality of service (QoS) in a way to make the delay within limits. Also, using multiple keys for encryption and decryption will make the probability of breaking the encrypted data is very difficult because even though if the first key has been known by a hacker, the probability of knowing the other key is very rare. In this research, it is very important to consider QoS metrics that are not going to be affected by the designed security system.

Cryptographic algorithms run on several kinds of networks and computing devices with different security features (KIZHVATOV, 2011). They include Enterprise servers and personal computers; also there are numerous embedded devices such as electronic keys, radiofrequency identification (RFID) tags, smart cards, mobile smartphones, and wireless access routers. Real examples of embedded devices along with their secure applications:

• Bank card with integrated chip, used to pay in a shop or to withdraw cash from an ATM;

• E-passport with an integral contactless chip;

• SIM cards that enable access to the cellular network and provides privacy of voice and data sent over the air;

• A Smart meter that records consumption of electric energy in houses and sends the results to the central energy office (KIZHVATOV, 2011).

In addition to the previous usage of security involvement, there are other areas that are important for security implementation such as industry and manufacturing. For instance, current manufacturing technology relies on big data and information technology (Khan, et al., 2017). Furthermore, the authors in (Butt, et al., 2018) state that there is a big relation between the technology and industry, so it is essential to enforce some security restrictions to safeguard the processing data.

1.2 Gap in Knowledge

The main concerns in the security of the network are: "current encryption algorithms consume a lot of resources such as, time and energy which are limited in a wireless network" (Ahmad, et al., 2016). The biggest challenge in a wireless network is how to get a good tradeoff solution between security and QoS (Bansod, et al., 2015). For instance, the real-time application is sensitive to the delay, and the energy of the battery is limited in the wireless device, but at the same time, the security of the information is urgently required as well, and nobody can give it up. So it is very important to balance these requirements.

Fig. 1-2 Balance of Security and Energy

Let us take this real-life example: Many people exchange voice messages through the Internet using their mobile devices, and most of them want these massages to be secret and nobody can listen to them. Yes, this is possible and can be done by encryption. For instance, the important (secret) voice msg in the WhatsApp can be encrypted by a strong cryptography algorithm like AES. But, if this msg is long, it will consume more mobile energy. And with many messages may consume all the battery and then switch the mobile device off.

Another example: sometimes we need to secure the routing protocol in WMANET to prevent the routing attacks. And this can be done by encrypting the request-response messages between the nodes. However, this encryption may cause a bit of delay in this protocol and may affect the network function and their QoS, in addition, to consume more energy from the participated nodes.

Here the gaps can be summarized as follow:

- Modern multimedia communication tools must have high security, high availability and high quality of service (QoS) (Liang & Han, 2011).
- Any security implementation will directly impact on QoS and power consumption (Amine, et al., 2018).

- Current encryption techniques are not suiting wireless real-time traffic in terms of delay, throughput and power consumption (Ahmad, et al., 2016), (Hamada & Rahman, 2016). And In the previous studies, there is no consideration for saving power by the wireless nodes during the encryption process.

- In the substitution transformation function, using single typical S-box alone does not have enough cryptographic strength for Advanced Encryption Standard algorithm AES, and it is vulnerable to cryptanalysis attacks (Ali, et al., June 2014).

- MixColumn function in AES algorithm account as the most expensive operation and consume a big amount of energy. So it is vital to create or modify a lightweight and low-cost encryption algorithm to solve the attacks in the wireless network for better complexity and protection (Masoud, et al., Sep 2015).

- The weakness of AES is that it works with a single key. For instance, a man in the middle attack can fraudulently capture the cryptography key and using it to reveal the encrypted data. Also, a brute force attack is still possible with several supercomputers.

- There is a lack of convincing security analysis for currently proposed cryptography algorithms.

The problems can be explained in details according to the literature:

In a mobile ad hoc network, the new nodes can join the authentication process where each node can send a request to the nearest servers in the local neighbourhood region. However, this makes the network vulnerable to Sybil attack (Eissa , et al., 2011) or packet sniffing.

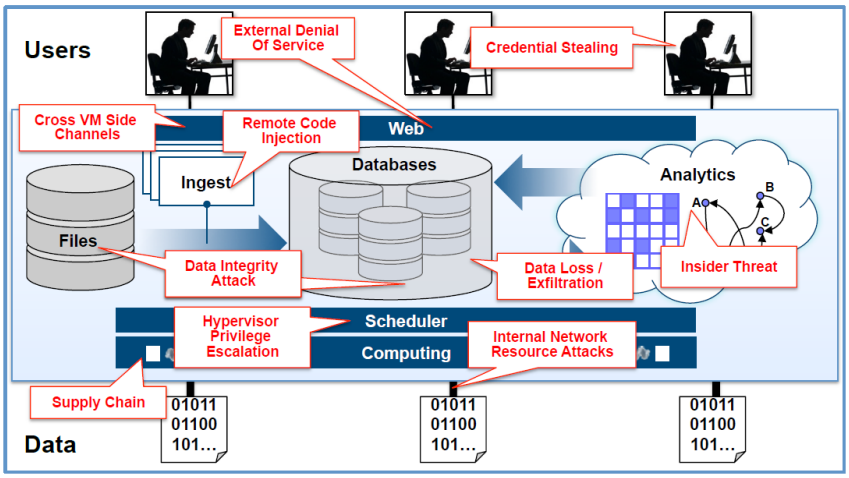


Fig. 1-3 Security Threat (Kepner, et al., 2015)

Many security challenges could be found in such an IoT data processing system (fig 1-3): external denial of service, credential stealing, cross virtual machine (VM) side channels, data loss, insider threats, internal attacks, and supply chain attacks (Kepner, et al., 2015). These attacks threaten to damage the availability of the IoT handling system, compromise the confidentiality of the data, and disrupt the integrity of the original data.

The encryption in the wireless environment is completely different from other environments. For instance, the encrypted data to be stored are don't care about the processing time or Energy consumption, when the data to be sent need quickly processing and may need less power to be consumed if it is being sent from a wireless device.

The encryption and decryption process consumes a significant amount of computing resources such as CPU time, throughput, and battery power (Jiehong & Detchenkov, 2016; RAMESH & UMARANI, 2012). According to (Sahu & Kushwaha, 2014) who state that after encryptions of a 5 MB file using Triple-DES the remaining battery power is 45% and subsequent encryptions are not possible due to battery dies rapidly. Also, AES showed poor performance results compared to other algorithms since it requires more processing power. Regarding voice concern, the results of (Samad , et al., July 2017) show that although IPSec can add security, it can reduce the VoIP performance in terms of higher delay and higher CPU usage. For these reasons, it is crucial to balance these requirements and achieve the best tradeoff solution for them.

1.3 Research Questions

To investigate and address the gap in the knowledge problem, the main question is:

> "**How can the security cost (delay, energy) be reduced without affecting the security level of cryptography implementation for voice over wireless devices?**"

This question leads to other sub-questions:

- What is the strongest encryption algorithm and what is its cost?
- Can the proposed encryption technique achieve an optimal trade-off solution between the security and performance compared with the standard AES?

So, the answer to these questions will fill the gap in the knowledge and achieve the aim of this research, to propose a lightweight encryption algorithm with the same level of security compared with the standard algorithm.

1.4 Research Objectives:

The research question and the issues highlighted in previous sections would be addressed in the following objectives to achieve the aim of this research:

Objective #1 is a critical investigation into the wireless network traffic and standard cryptography algorithms.

Objective #2 investigates and studies the network traffic and their requirements. Also, it experimentally investigates the AES encryption algorithm.

Objective #3 Implementation of multi S-box encryption algorithm with a high level of complexity and at the same time keeping the execution time and power consumption at the same level. Moreover, address the fixed structure of SubByte transformation function, to increase the confusion in the cipher.

Objective #4 to design, implement, and build a lightweight and low-Energy encryption algorithm for audio files considering the cryptosystem strength. To meet the wireless devices requirements (limitation)

Objective #5 conceptual development of an innovative encryption algorithm with triple key and high level of complexity with effective execution costs such as time and energy consumption. Also, to Validate, Evaluate and Analyses the new security architecture in every step of the design to prove their strength.

These objectives will be justified in the methodology chapter; also, they will be explained in details in each related chapter.

1.5 Knowledge gap filling Contributions

The goal of the research is to reduce the execution time and power consumption of the encryption process compared with the standard algorithm and at the same time at least to maintain or to increase its security level, to get high performance with high security.

Fig. 1-4 Metrics Enhancement

This goal has been achieved, and this research contributed to the knowledge by:

Contribution #1 Study and Investigation of the security and encryption techniques for wireless environment.

Contribution #2 Design, Implement and Analysis of cost effective Lightweight Encryption Algorithm for Audio files.

Contribution #3 Propose a Novel Triple Key Encryption Algorithm (TKE) with high complexity and lightweight characteristics.

The published works of these contributions are stated in fig. (1-4), in addition to the relate chapters and objectives. The figure illustrates the relation between the objectives and the chapters to achieve the contributions and the outcome publication for each contribution.

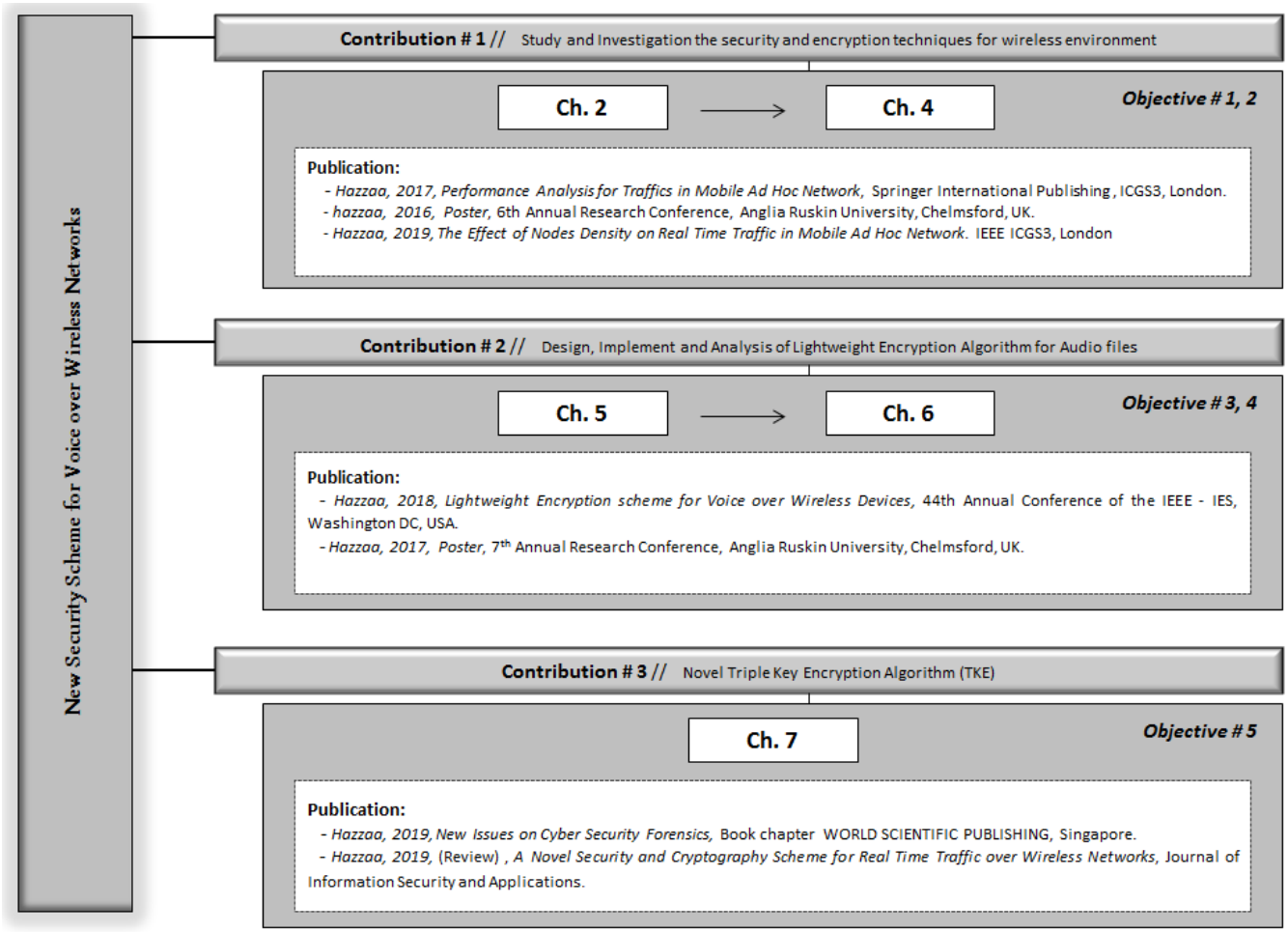


Fig. 1-5 Research Contributions

1.6 Thesis Structure

Chapter 1- Introduction

This chapter will introduce the field wireless network security and challenges that affect the privacy and information of users. This chapter presents the cryptosystem and their relation to data security in addition to its implementation cost. This chapter has included research questions, objectives, and knowledge gap filling contribution of the research. A brief description of each chapter in this thesis is included.

Chapter 2- Literature Review

This chapter gives a review for the literature and an overview of the background from more than 45 research papers that have been published in the area of Wireless networks security and cryptography. It explains the network types and their infrastructure and the topology and then explains their challenges and security concerns. Furthermore, the real-time application and their requirements have been reviewed as well. The chapter has in detail explained the security requirements, implementation, and limitation in wireless environments in addition to the parameters measured. The gaps of knowledge that are highlighted by previous researchers are summarised and became the basis for the motivation and aims of this research.

Chapter 3 – Research Methodology & Proposed Framework

This chapter describes the approach that was taken for the proposed design and its elements. Justification of research method and rationale for the research approach are explained. The details of how the results tested and analysed are also included. Furthermore, the whole research framework is graphically illustrated in this chapter.

Chapter 4 – Study and Investigation

This chapter investigates and studies the network traffic and their requirements. Also, it experimentally investigates the AES encryption. The aim is to find the most sensitive traffic to the delay and quality of service metrics, also to determine what the strongest cryptography algorithm is. Many scenarios will be carried out in a wireless environment to test the different traffic type over the wireless network, and with a variable number of the nodes.

Chapter 5 – Multi S-box Encryption Algorithm

This chapter is going to investigate and develop an encryption algorithm which aims to increase the complexity of encryption process making it difficult to breach and at the same time does not increase the execution time and power consumption. The base of the work is the AES algorithm functions. A SubByte function using multi S-box transformation technique has been suggested to increase the confusion and complexity of encryption algorithm and to make it ready for further enhancements in the following chapters.

Chapter 6- Lightweight and Low-Energy Encryption Scheme

In this chapter lightweight and low energy encryption algorithm for voice over wireless networks is being developed and tested. The new encryption algorithm has to meet the QoS requirements of voice traffic and to be suitable for wireless devices. The aim of the chapter is to reduce the execution time and energy consumption of the encryption process compared with the standard algorithm (AES) and at the same time at least maintains or increases its security level.

Chapter 7 - A Novel Triple Key Encryption Algorithm (TKE)

This chapter will develop the algorithm that has already been proposed in chapter 5 and 6, to increase the security level by adding 3rd key function. The SubByte function proposed in chapter 5 and mixcol proposed in chapter 6, in addition, the 3rd key function will be all used to propose the novel encryption algorithm for high security and lightweight consumption. A huge performance and security analysis have been conducted in this chapter to demonstrate the effectiveness of the novel design.

Chapter 8 – Conclusions and Future Research

A summary of each chapter conclusions is presented in addition to the major achievements and detailed recommendations to the wireless network designers. A further research topic also explained and will be required to continue developing other cryptography schemes to meet the new developing in wireless network’s needs.

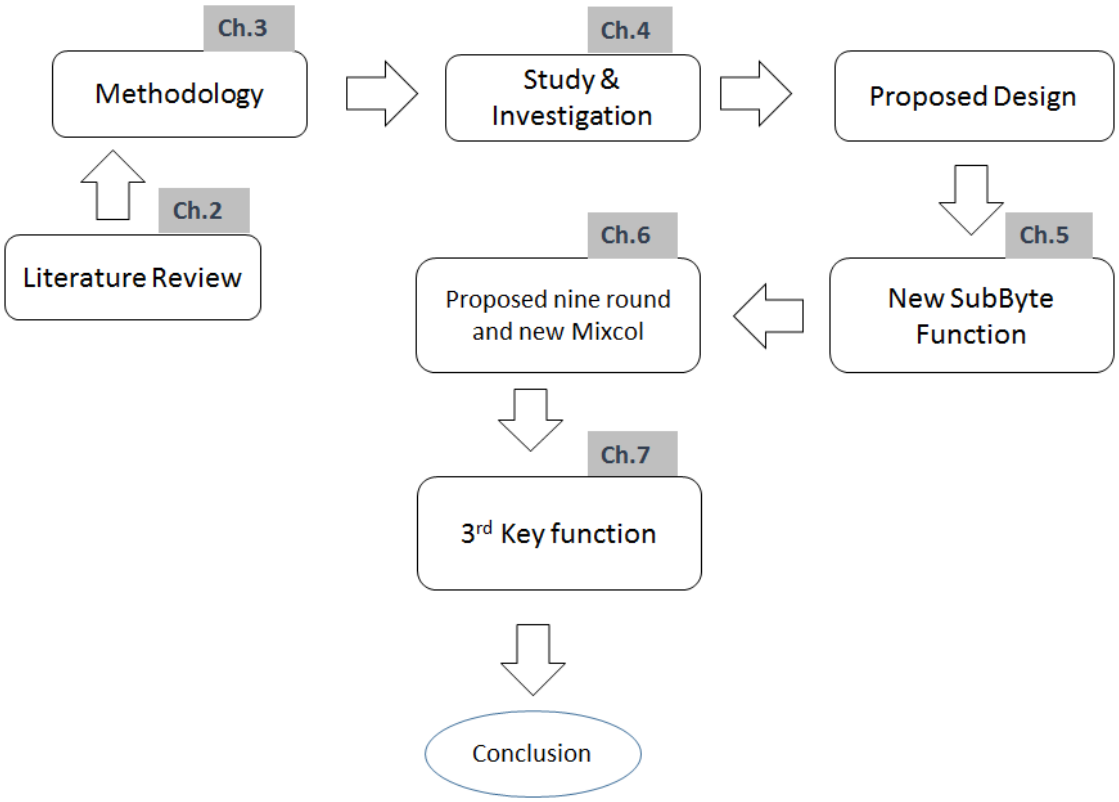


Fig. 1-6 Chapters Structure

2 Chapter Two: Background & Literature review

2.1 Introduction

This chapter gives a review of the literature and an overview of the background for network security and cryptography. First, it will explain the network types and their infrastructure and the topology and then explain their challenges and security concerns. The limitation of network elements has also discussed in this chapter. Furthermore, the real-time application and their requirements have been reviewed as well. The chapter has in detail explained the security requirements, implementation, and limitation in wireless environments and the main parameters. The focus on cryptography in this chapter takes big attention because it is the main objective of this research. Also, the implementation of encryption algorithms in computer networks has discussed and their advantages. A quick overview of parallel computing has also been mentioned in this chapter. Finally, we discussed the related work at the end of this chapter. The advantages of this chapter will help to understand the main problems in the wireless network and the security concerns and voice traffic issues. It will help to diagnosis the gaps in the network security and to put the main research questions and to choose the main objectives to address these gaps. The following sections explain in details the chapter parts.

2.2 Computer Networks

Computer/Internet network is a set of computers/nodes connected together for the purpose of sharing resources and data. In computer networks, computing devices exchange data with each other using connections between nodes. The internet has been developed over the last 45 years (Thomas & Robertazzi, 2017)

2.2.1 Networks Types and Topology

There are two types of network:

- Wired Network: is a set of connected PCs through the wired link. Most of the wired networks use Ethernet cables to transfer information throughout the connected nodes. These networks may use the routers or switches to organize the network depending on the size of it (Cisco, 2016).

- Wireless Network: is a set of connected devices that are not connected by cables of any kind. These networks use the access points to get access to the network or use some wireless technologies such as ad hoc and Bluetooth (Zheng, et al., 2013). These networks usually are cost-effective and easy deployed.

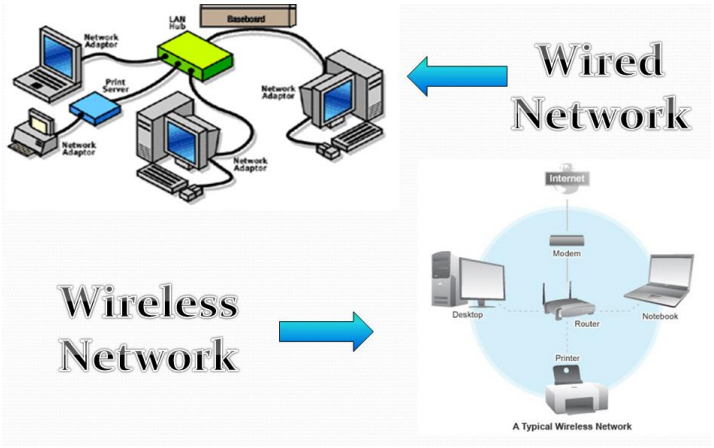


Fig. 2-1 Network Types

Also, there are two kinds of topology can connect the networks; that is, infrastructure and infrastructure-less network (Jahankhani, et al., 2017).

2.2.1.1 Ad-Hoc Network

An ad-hoc network is an infrastructure-less network. It is a crucial enabler of next-generation communications. Such networks can be formed and reconfigured dynamically and they can be mobile, standalone or internetworked with other networks (Panaousis, 2012), for instance, Mobile ad-hoc network (MANET) is the most important part of these networks.

Mobile ad-hoc network (MANET) is a set of mobile devices that communicate with radio links. MANET network infrastructure is not defined and there is no centralized administration for controlling the other activities (Hazzaa & Yousef, 2017; Aarti & Tyagi, 2013; Jahankhani, et al., 2017). Mobile ad-hoc network MANET is established by a set of independent devices/nodes that connect with each other by creating a multi-hop radio network and retain connectivity in an infrastructure less method. It allows data, voice and video communications over a wireless channel (Thomas, et al., 2010). The main role of MANETs is to assist wireless and mobile communication services without using the costly service provider network and without having a previously set up infrastructure.

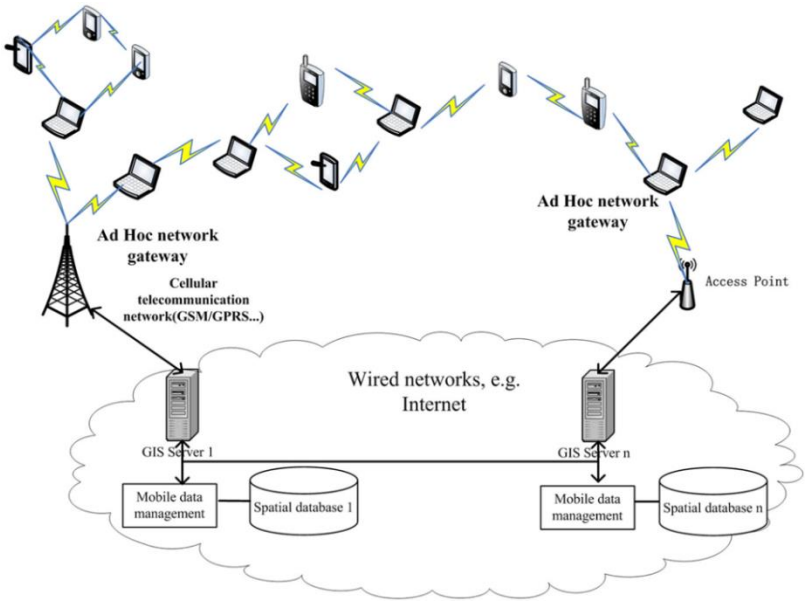


Fig. 2-2 Ad Hoc Network

Security is essential in wireless ad-hoc networks, and mobile ad-hoc networks (MANET) face a major problem in terms of their lack of central control. There are numerous internal and external sources of potential attacks on networks, requiring tailored protection for applications. For instance, Vehicular Ad-hoc Network (VANET) is used to wirelessly connect vehicles, utilizing mobile vehicles as nodes of communication to locate and detect movement among vehicles linked into the system. Data packets circulating in wireless networks such as VANET, where speed, optimized geographical coverage and streamlined data processing are overriding specification concerns, are generally validated without assessment of content accuracy or source quality. This renders such systems susceptible to erroneous and potentially malicious messages being disseminated (e.g. by spoofing), such as attacks to cause denial-of-service (DoS) (Sharma , et al., 2017). Consequently, there is an urgent imperative to increase the security of such networks.

2.2.1.2 Wireless Sensor Networks

Nowadays, the world is rapidly moving toward the wireless environment. In 2020 most of the devices will connect wirelessly. And the most significant part of this technology is wireless sensor networks WSN. According to (Matin & Islam, Sep.2012), WSN can be defined as " *a self-configured and infrastructure-less wireless networks to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants and to cooperatively pass their data through the network to a main location or sink where the data can be observed and analysed*".

Currently, many researchers are focusing on Real-time Wireless Multimedia Sensor Networks (WMSNs), Because it supports many application layer services to solve real-life concerns, such as health monitoring, transport management, weather forecasting, and safety & security investigation (Elhoseny & Hassanien, 2018; Ehsan & Hamdaoui, Jun 2012). A traditional WMSN contains a range of sensor nodes, including both simple and multimedia sensor nodes, and a Base Station (BS) as shown in Fig. (2-4)

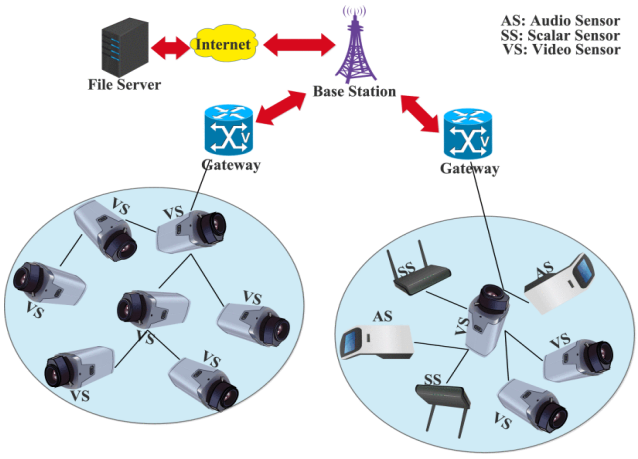


Fig. 2-3 Traditional wireless multimedia sensor network (Usman, et al., 2018)

There are several advanced technologies such as the Internet of Things and IPv6 over Low power Wireless Personal Area Networks (6LowPAN) have enabled WMSNs to carry out the remote control and surveillance using the Internet (Usman, et al., 2018). Similar to MANET, wireless sensor network has huge concerns and limitation about security implementation. Also, the nature of this network and the node sizes are affecting the network function especially with real-time traffic like voice (more details of these issues will be explained in the next sections of this chapter). Extensive research is required to improve technology in this area, particularly given the importance of WSN applications (e.g. for monitoring in industrial, environmental, and military contexts) (Yick, et al., 2008). Solutions generally seek to devise novel services, algorithms, protocols, and designs to enable diverse applications.

2.2.2 Real-Time Traffic in Network

2.2.2.1 Basics of Audio Files

Audio technologies generally seek to convey (i.e. reproduce) sound patterns at a remote location, possibly at a later time (Thomas & Robertazzi, 2017; Watkinson, 2001). Sound

comprises physical vibrations and rapid fluctuations in pressure (i.e. multiple times per second), usually heard through the air. Pressure cycles affect the frequency received (i.e. heard), this frequency and amplitude are the main parameters of sound (Waggoner, 2010). Real sound waves comprise continuous analogue values, which can be of any frequency and amplitude (Waggoner, 2010); humans can optimally hear sounds in the frequency range 20-22,000 Hz (Salomon, 2012). Prior to the digital recording of sound, discrete representation was required to store values of continuous signal amplitude in computers (Marina, 2009).

2.2.2.2 Digital Audio

Analog and digital audio recorders have similar features, but the latter is distinguished by improved channel number, and sampling size and rate. In order to apprehend sound's spatial characteristics, simultaneous recording in varying locations is necessary (Massa, 2000).

Various methods exist to convert sound from analogue to digital formats (Jahankhani, et al., 2017), including Pulse Code Modulation (PCM), whose use is prolific, using digital audio, as shown in Figure (2-5) The x-axis (time) is not represented as continuous, rather it is in stepwise or discrete form, with waveform being noted at regular points; the effectiveness of this longitudinal sampling technique is dependent on the frequency of sampling (sampling rate), Fs, which is generally predetermined and static (i.e. unresponsive to varying signal frequencies). Accuracy can be diminished by factors such as jitter in the sampling clock, and any error in the time base changes the moments when sound samples are recorded, and this effect is identifiable. Digital audio systems treat irregular time base by temporary memory storage of samples in a stable, known as time base correction, which totally eliminates errors (Watkinson, 2001)

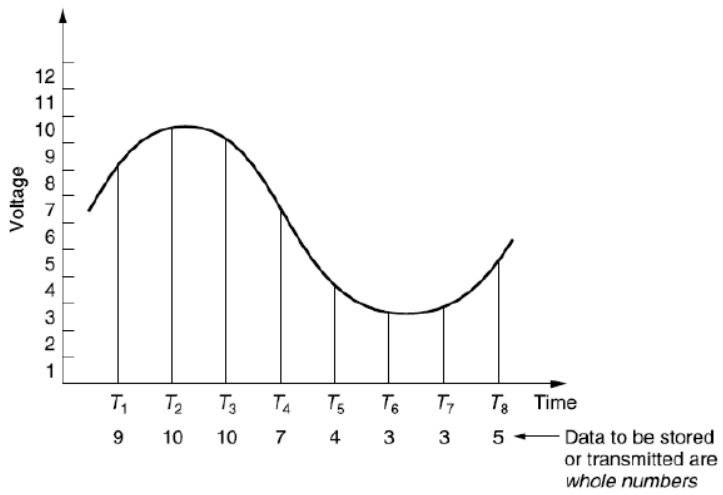


Fig. 2-4 Pulse Code Modulation (PCM) (Watkinson, 2001)

Audio files that are known and played on a personal computer come in different formats which are usually associated with different file extensions such as (*.wav) format and (*.mp3) format (Balagurusamy, 2009).

2.2.2.3 WAVE Audio Format

Waveform audio format, known as WAVE, is commonly used in Windows, which with most other operating systems supports audio files in native formats, in addition to video and image files. WAVE files are aggregates of data chunks, which begin with headers (descriptor chunk) and might include sub-chunks (Figure 2-6). Format chunks include specifications that determine the waveform, including bits per sample, byte rate, and sample rate. The size of sound data and raw data is indicated by the data chunk, and the potential future addition of novel chunk types may not be recognized by existing software, in which case such chunks are generally skipped during data processing (Massa, 2000).

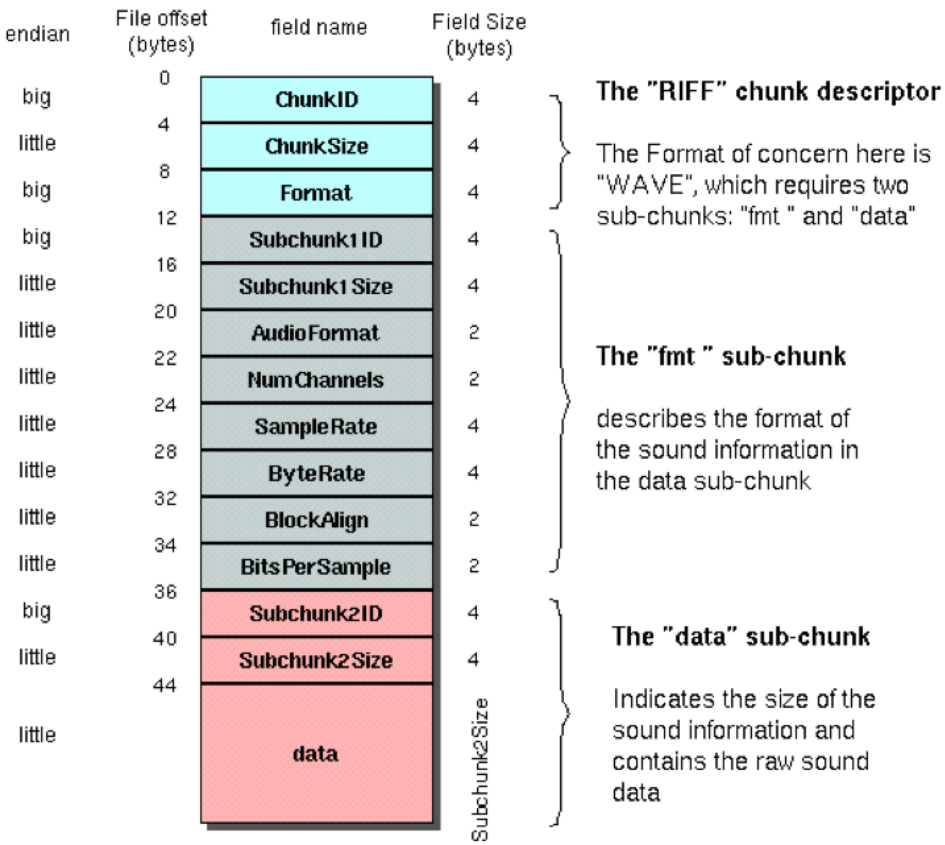


Fig. 2-5 WAVE file layout (Bhalshankar & Gulve, 2015)

2.2.2.4 Voice over Wireless Networks

The voice requirements in wireless networks are different from wired networks. In the wireless networks, we must carefully investigate and analyse the problems associated with this integration (Thomas & Robertazzi, 2017). A real-time application like the voice has a sensitive requirement in term of delay. For example, in phone call conversation, the voice should reach the destination quickly and efficiently. Any delay in the voice will affect the quality of the call. This problem will also be affected if apply security rules. Furthermore, this problem will become worst in the wireless network because of the nature of wireless networks. Many researchers gave good attention to real-time application in wireless networks. They tried to address some issues concerning security and delay. For instance, (Binod & Hyuk, 2010; Emmanouis & Christos 2012) suggested a secure model for autonomous networks like mobile ad hoc networks to create real-time communication in disaster rescue operations. Secure P2PSIP, intrusion detection, and secure routing methods are a vital part of any secure model. The design of these extensions helps to meet the requirements of a mission-critical MANET where rescuers should create links to communicate with each other by using lightweight devices such as PDAs. A novel security framework was designed by Liang and Han (2011) to enable different multimedia services in the IoT environment. Also, they presented a novel media-aware traffic security style based on the proposed traffic classification to enable various multimedia services supplying to customers anywhere at any time. Furthermore, they proposed the scheme basis and approach to performing a good trade-off between system flexibility and efficiency.

However, some problems remain in cryptography and security schemes for real-time applications, including the security of voice in wireless networks while maintaining quality, given the heavy demand for high-quality data and the absolute necessity of high-speed communication in real voice applications So, the question is: how to secure the voice in wireless network without affecting their quality? As mentioned, the voice needs speed and security at the same time, and this should be addressed in this research.

2.2.3 Power in Wireless Devices

The potential of wireless networks has been evident since the 1970s and in more recent years numerous specialty networks have emerged, including wireless personal area networks (WPANs), wireless local area network (WLANs), wireless metropolitan area networks (WMANs), and wide area networks (WWANs) (Abhijith, Srivastava, & Mishra, 2013). All wireless networks have high power loads, and each node requires efficient management,

which pertains to the field of "Green Computing", although the main interest is in cutting power costs rather than reducing environmental impacts. In relation to cryptography and security in general, the addition of security processes adds an extra burden on the energy topography of the system. Empirical studies have demonstrated the high power cost of implementing encryption algorithms like AES, Bluefish, DES, and RC6 (Masoud, Jannoud, & Ahmad, Sep 2015).

Consequently, the negative impact on power consumption inhibits the use of cryptographic algorithms in smartphone communication applications. Consequently, power consumption is a real and pertinent issue in encryption algorithms, which must consider such real-world consumer concerns in addition to pure cryptographic and security performance. There is clearly a need to provide effective security for wireless networks without increasing node power consumption inordinately. So, the question is: how to secure the data in a wireless network without affecting the power of the nodes? As mentioned, the wireless nodes need to keep their power as long as possible when implementing the security enforcement and this has been addressed in this research.

2.2.4 Networks Security Challenges

The recent increase in wireless portable users and data usage (Thomas & Robertazzi, 2017) (Butt, et al., 2018), has added more load on networks. Real-time applications over wireless networks often suffer from jitter, delay, and packet loss. In future smart cities, there will be a solid placement of wireless sensor networks WSN ranging from reserving water to public safety. "*These WSN and Things Internet (IoT) should be seamlessly integrated with future 5G networks. In cellular networks such as 3G, 4G, and LTE, mobile devices are served by Base Stations (BSs) which cover a large area (about 1-2 miles). Due to many practical deployment issues, some areas have good coverage while other areas may not. As a result, the wireless signal strength of a mobile device varies based on its location*" (Hu & Cao, March 2017) said. IoT devices are an attractive attack target for cybercriminals. Internet of Things (IoT) devices frequently employs weak security actions, and their compromise could lead to safety threats and privacy breaches in the real world. Lately, some malware families were created to target vulnerable IoT devices (Albonda, Tapaswi, Yousef, & Cole, March 2017) (e.g., routers, IP cameras, and CCTV) and form botnets for DDoS. It is estimated that some IoT botnets comprise more than one million infected devices. In October 2016, the botnet attacked the DNS service provider, taking down a large portion of websites in North America, including GitHub, Twitter, Netflix, and so on (Cheng, et al., 2017). Also, the

privacy of information, transferring into the network, is repeatedly threatened by hackers. For example, a man in the middle attack can reveal secure information by deceit the users of the network.

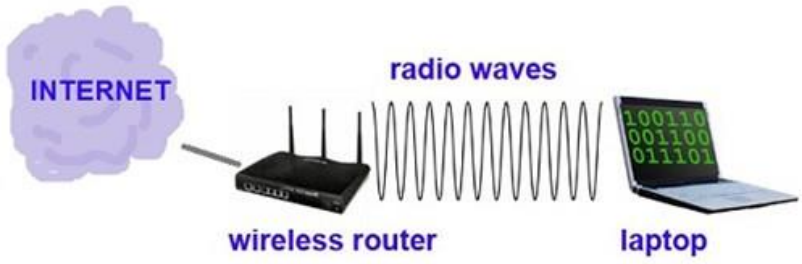


Fig. 2-6 (purevpn, 2017)

2.3 Cybersecurity

Cybersecurity is the processes and technologies designed to protect computers, networks, data and programs from unauthorized access, damage or attack (Cisco, 2018). According to the Telegraph in 2017, the cost of cyber-attacks which have spent by UK businesses reached $34 billion. Ensuring cybersecurity needs corresponding efforts throughout an information system. The main elements of cybersecurity include Applications security, Information security, Network security, Disaster recovery and End-user education (Jahankhani, et al., 2017). The fig. below shows the global cyber security market size from 2014 to 2024.

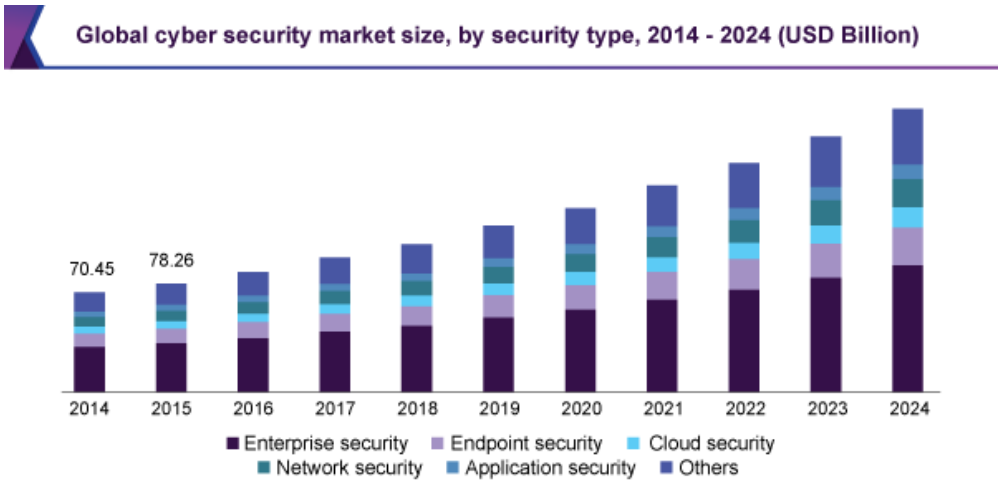


Fig. 2-7 Source (Market.Research, 2018)

According to (Rouse, 2014), "*Application security is the usage of hardware, software, and technical methods to protect applications from external threats. Arrangements taken to*

guarantee application security are sometimes called countermeasures". The most common hardware countermeasure is a router that can prevent the IP address of an individual computer from being directly visible on the Internet (Bhaskar & Ahson, 2008). While the simplest countermeasure in terms of software is a firewall application to limit data handling or files execution by specified programs installed in the computer. Mainstream countermeasures also include biometric authentication, antivirus, and anti-spyware programs, conventional firewalls, and encryption/decryption programs.

2.3.1 Security Principles

A brief review of the main requirements that all networks usually have to accomplish would be explained in this section.

1. **Confidentiality** means that secret information should not read or revealed by others, except the authorized persons (Goodrich & Tamassia, 2011). So to achieve this principle, there are many tools could be used. These tools incorporate the following concepts:
    - Encryption: the changing of information in a way that cannot be easily read or understood. This can be happened by using some known algorithms with a secret key.
    - Access control: a set of rules that limit access to privet data.
    - Authentication: it means that the users or system should give its identity by using different methods, like a password or a smart card.
    - Authorization: the determination if a person or system is allowed access to the network, depends on access control rules.

    Confidentiality guarantees that message content is never revealed to WMANET entities that are not authorized to interpret it. Due to WMANETs' wireless links being easily susceptible to eavesdropping, confidentiality is very crucial for protecting the transmission of private information (Stallings, 2017). Especially, any leakage of data or control traffic (such as routing) information could be really harmful in certain circumstances such as emergency cases, where human life is in danger. Ransomware insertion is nowadays the most popular attacking vector because it denies the availability of critical files and systems until attackers receive the demanded ransom (Jaime , et al., 2019). In this case, any malicious MANET node is likely to try revealing the confidential message content as the first step towards different kind of physical or network attacks. (Apietro , et al., 2014).

2. **Integrity** means that secret information should not be altered. And the malicious modification to the information which traveled in the network must be (quickly) detected.
3. **Availability** The services provided by the network must be always available, despite any faulty of the system.
4. **Non-Repudiation** assures that the sender of data could not deny having sent it or the receiver could not deny for received it.

2.3.2 Threats and Attacks

- **Routing Attacks**: the routing signal is modified by unauthorized nodes and rerouted in an incorrect route in order to provide unauthorized data access to the attacker. Such attacks include breaking the neighbour, black hole, routing table poisoning, wormhole, and replay attacks.
- **Eavesdropping:** is a passive attack that could not be identified, because it has no effect on the process of the routing protocol (Goodrich & Tamassia, 2011). The aim of the attacker is to gain access to the data and try to break the encryption of the data.

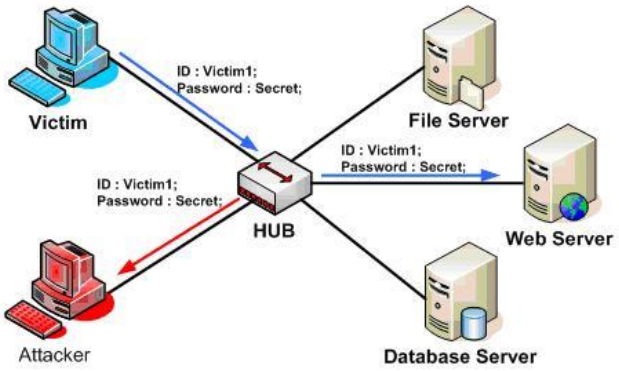


Fig. 2-8 Eavesdropping (SSL, 2015)

- **Denial of Service:** this attack aims to disrupt routing in order to prevent the network from functioning and providing normal service. The attacker sends a massive volume of simultaneous requests to overwhelm the server and hamper its response time by jamming its mechanisms, ideally halting the complete system.

- **Sybil Attack:** is an attack against the wireless sensor networks where several frank identities with fake identities are used for getting an illegitimate entry into a network (Suriya & Rajamani, 2015). Basically, a Sybil attack means a node which pretends its identity to other participated nodes.

- **Masquerading:** During the neighbour acquisition process an attacker might act as another node, capturing messages and replaying or modifying them posing as a legitimate node.
- **Alteration:** The attacks on integrity; when someone makes unauthorized changes to codes or data (Northcutt, 2018).

2.3.3 Security Issues

This section gives an overview of security issues, particularly in wireless networks. Modern wireless networks, such as in 5G environments, comprise core networks based on IP, and include numerous service providers and wireless technologies, with the likelihood of mobile internet devices switching between different technologies and providers to maintain optimum QoS. Rapid vertical handover and network accessibility and openness render devices vulnerable in some respects, including availability, privacy, communication security, access control, and data confidentiality, in addition to IP-related weaknesses (Amine, et al., 2018) Clearly 5G networks consequently raise numerous new privacy and security challenges (Panwara, et al., 2016).

Furthermore, in addition to dynamic vertical handover, 5G mobile devices are ubiquitously connected (i.e. they are permanently networked at all times) to ensure continuity of service and enable background updates and notifications. Obviously increased (i.e. permanent) connection offers the maximum window for malicious attacks like impersonation, eavesdropping, man-in-the-middle, denial of service, replay and repudiation attack (Ferrag , et al., 2017). "*Maintaining a high level of QoS in terms of delay, when huge volume of data is transferred inside a 5G network, while keeping on the same time high security and privacy level, is critical in order to prevent malicious files from penetrating the system and propagating fast among mobile devices*" (Amine, et al., 2018). Consequently, the challenge of maintaining QoS by minimizing delay while applying necessary security measures is increased in 5G, and it is incredibly difficult to provide rapid, high-quality communication with the requirement of zero latency (Basaras , et al., 2016). As mentioned, eavesdropping is the most dangerous attack because it reveals the information and breaks the confidentiality of the data and security. The best solution for this attack is encryption. However, the encryption should be lightweight and consider wireless environments limitations.

2.3.4 Applications of Encryption in Network Security

In addition to data encryption, there are many uses of cryptosystems in the network environment. This section explains some examples of network security and shows the importance of using encryption in Internet and network security and privacy. This section explains the issues involved in successfully incorporating IPsec encryption into VOIP services. According to National Institute of Standards and Technology (NIST, April 2004); Firewalls, gateways, and other such devices can help keep intruders from compromising a network, but firewalls are no defence against an internal hacker. They state that a need for another level of protection is required at the protocol stage to defend the data itself. In voice over Internet protocol VOIP, this can be carried out by encrypting the packets at the IP layer using IPsec. However, Cryptographic operation itself may cause an extreme amount of latency in the VOIP packet delivery. This leads to corruption of the voice quality, so once again there is a trade-off between voice quality and security, and a necessity for speed.

2.3.4.1 IP Sec

The network protocol suite IPsec encrypts and authenticates data packets sent over a network. It is essential for data privacy. It deploys security services over the internet protocol to cryptographically protect communications in VPN tunnelling (Figure 2.9).

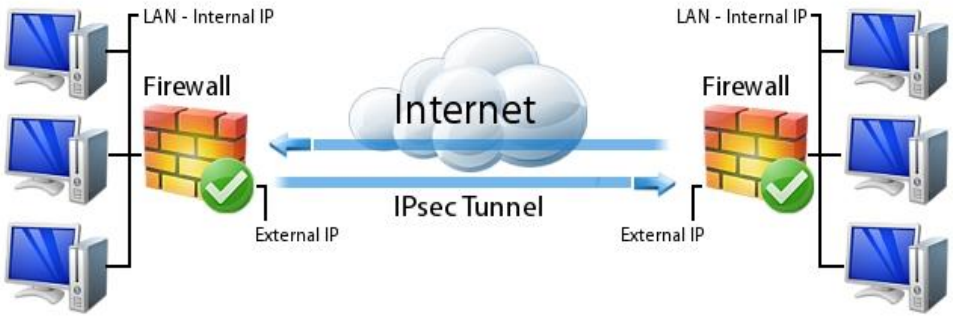


Fig. 2-9 IPsec (Techbast, 2016)

Authentication Header (AH) and Encapsulating Security Payload (ESP) IPsec protocols confer source authentication, connectionless integrity, and anti-replay service (Blaze, et al., 2001). Encrypting the IP address of the origin prevents attacks on the packet (i.e. by hackers), and precludes them from determining the packet originality. However, IPsec has high latency, particularly for voice, with higher CPU usage in VoIP (Samad , et al., July 2017).

IPsec supports transport and tunnel delivery modes (Figure 2.10). Upper layer headers and payload (data) are encrypted in the IP packet in transport mode. Normal view is used for the

new IPsec header and the IP header, thus the interception of an IPsec data packet would indicate to attackers the target destination of the packet, enabling traffic analysis, but would not reveal the content. In tunnel mode, the whole IP datagram is encrypted and enclosed in a novel IP packet, with the encryption of the IP header and the payload; the only clear information is the new IP Header and the IPsec header for the encapsulating packet. Each "tunnel" is usually shown between router and gateway network components, or for mobile users, between a client and a router/gateway.

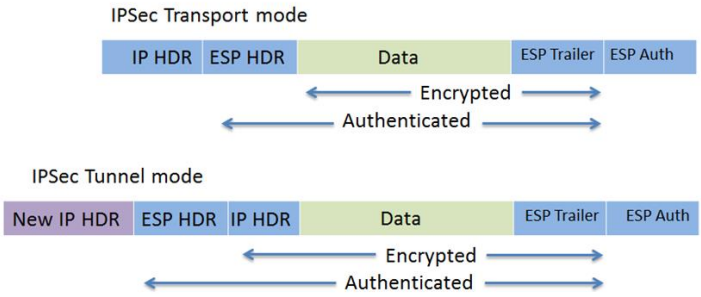


Fig. 2-10 Daniels Networking Blog 2017

VoIP data can be dynamically protected by IPsec encryption as it passes through the network, preventing deciphering of encrypted contents by intercepting attackers (e.g. if they override physical precautions for network security). Compared to traditional landlines, IPsec enables more secure VoIP communications, but despite its popularity, it incurs overheads (Kolahi, et al., July,2017), consequently, it increases delay and increases retransmission attempts, reducing throughput (Hamada & Rahman, 2016) However, despite incurring some delay, the IPsec protocol is successful in its main aim of increasing network security, and research attention is devoted to improving and adapting encryption algorithms rather than devising alternatives.

2.3.4.2 SSL

"Secure Sockets Layer (SSL) SSL was a ubiquitous protocol before it was superseded by Transport Layer Security (TLS) in 1999 (Bahrak & Aref, July 2008), but the latter continues to be commonly referred to as 'SSL' by academics and industry practitioners (GLOBALSIGN, 2018). SSL provides safety to channels between devices over the internet or internal networks, such as between a web server and web browser, in which case 's' for 'secure' is suffixed to the website address 'HTTP' (i.e. 'https') (Amine, Maglaras, & Argy, 2018).

2.3.4.3 VPN

Comprising the services and technologies used by VPN providers to provide fast and secure connections to VPN servers, VPN protocol includes transmission protocols and encryption standards. An example of private networks encryption is shown in Figure 2.11.

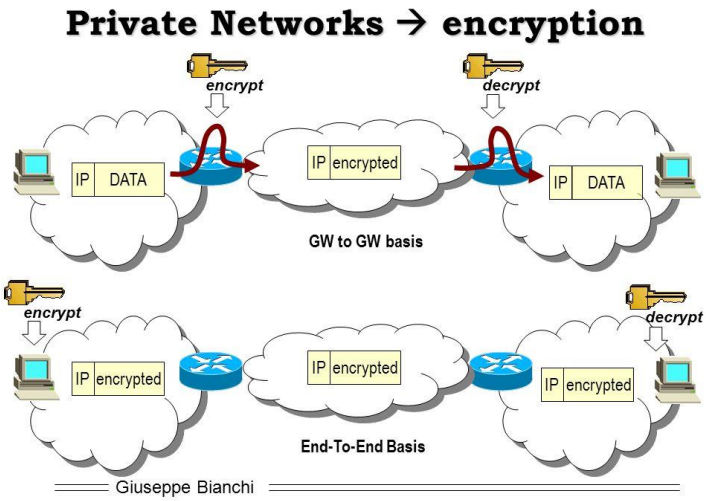


Fig. 2-11 VPN

There are many methods to mitigating security challenges. Data protections are mostly useful for preserving the secrecy of the information (Thomas & Robertazzi, 2017). Typical defences of this type include encrypting the links between users and the IoT processing system or between the data sources and the IoT treating organization or encrypting the information in the file system.

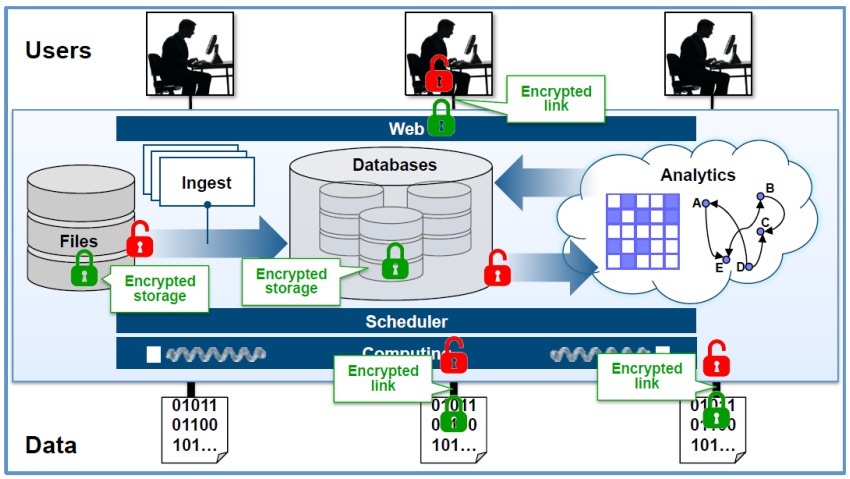


Fig. 2-12 Methods of Data Security

To summarise, Cryptography is the mainstay of security solutions for networks, but developing more sophisticated techniques and algorithms, etc. require addressing the fundamental trade-off problem between QoS requirements for maximum speed and quality and minimal power consumption and the provision of security solutions that inevitably entail overheads in these dimensions. So, this should be considered in this research.

2.4 Cryptography

Cryptography in the dictionary means *secret writing*, (Crypto: Secret, graph: writing). It is "*The discipline which embodies principles, means, and methods for the transformation of data to hide its information content, prevent its undetected modification, prevent its unauthorized use or a combination thereof. Cryptography deals with the transformation of ordinary text (plaintext) into coded form (ciphertext) by encryption and transformation of ciphertext into plaintext by decryption*" (Trapnell & French, 2016).

The easy example to illustrate the encryption can be explained here, let us say:

FIRAS STUDY IN ANGLIA RUSKIN

This msg can be encrypted using a key between the sender and receiver. Let us assume the key =3 so we can shift the letters three positions to the left so the output will be

ASSTU DYINA NG LIARUS KINFIR

Now the above sentence is unreadable. And nobody can understand it apart from the sender and receiver who know the key.

Cryptography is about constructing and analyzing protocols that prevent third parties or the public from reading private messages. The main target of using cryptography is to achieve confidentiality. And this can be achieved by encryption. The encryption consists of two parts: the encryption algorithm and a secret key. There are two types of encryption process (Hercigonja & gimnazija, 2016) :

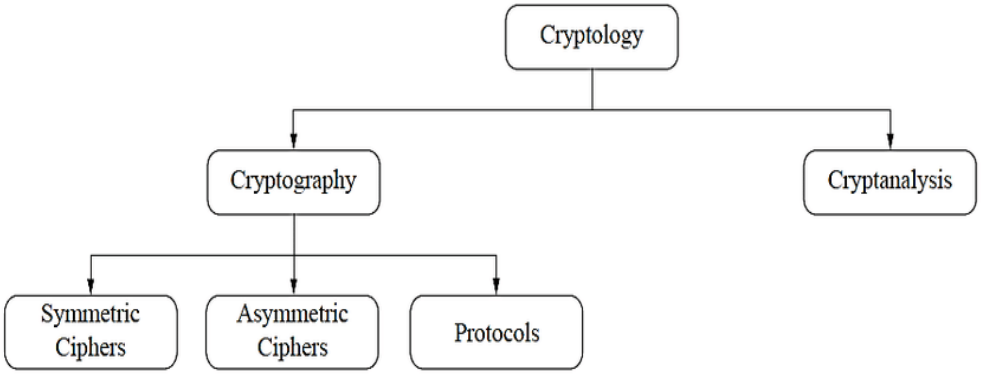


Fig. 2-13 Cryptosystem

- **Symmetric-key cryptography** refers to encryption methods in which both the sender and receiver share the same key.
- **Asymmetric-key cryptography** when each party has a different key, the public key, and private key.

The traditional asymmetric key algorithm may be slower than the symmetric key algorithm by 1000 times or more (Prakash, et al., Aug. 2015)

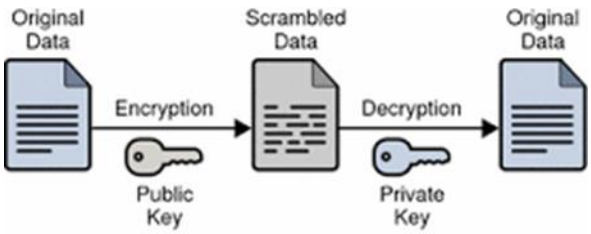


Fig. 2-14 Encryption Process

In regard to the efficient implementation of cryptographic algorithms, it has been focused on the major research efforts for the last two decades. Majority of cryptographic algorithms utilize arithmetic operations on finite mathematical structures such as finite multiplicative rings, groups, and finite fields (Savas & Koc, 2010).

2.4.1 Encryption Algorithms

In this section, a quick overview of some encryption algorithms is explained, which are used in the data security field and as following:

DES: Data Encryption Standard was the pioneering encryption solution (National Institute of Standards and Technology), first published in 1975. DES is 64 bits key size with 64 bits

block size. It became increasingly insecure as a block cipher as attackers developed more sophisticated techniques. (Umaparvathi, 2010). The most reason for its vulnerability is the small key size it has, which make it less resistance to a brute-force attack.

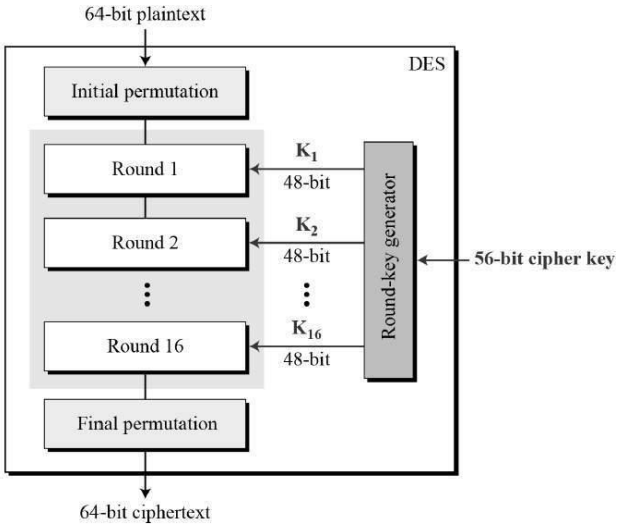


Fig. 2-15 DES (Website, 2018)

3DES: was an initial improvement of DES, revising the original 64-bit block size with 192 bits key size. It also applies DES thrice, to increase the level of encryption and typical safe time, which makes it slower than alternative block cipher methods (including DES, although it is more secure than this).

RC2: is a block cipher with a 64-bits block cipher and variable key size, from 8 to 128 bits. It uses 234 chosen plain texts, making it susceptible to related-key attack.

Blowfish: is another block cipher 64-bit block to replace DES. It uses a variable-length key, ranging from 32 bits to 448 bits, with a default of 128 bits. Blowfish is a license-free, unpatented, open-source package, with variants of up to 14 rounds (Ebrahim, 2013).

AES: is a block cipher with a variable key length of 128, 192, or 256 bits; the default is 256. It encrypts data blocks of 128 bits in 10, 12 and 14 rounds, according to key size. It is flexible and quick, and it can be used in numerous platforms, making it particularly useful for smaller devices (Ayyappadas, et al., 2014). It has frequently tested. In October 2000, NIST released the Rijndael algorithm for AES. Symmetric key AES algorithm became the most prevalent algorithm used to send information securely, and it is highly resistant to various attack types (NIST, 2004).

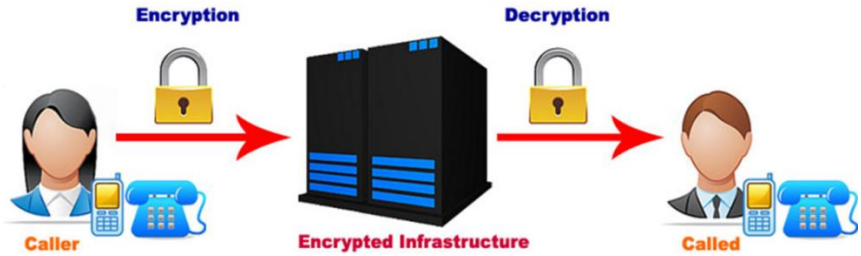


Fig. 2-16 Voice Protection

2.4.1.1 Security Analysis for Algorithms

According to the capability of attackers to break them, different algorithms confer varying levels of security. As with protecting systems, attacking network security involves a cost-benefit analysis. If it is prohibitively expensive to break algorithms (relative to the value of targeted, encrypted data), in terms of the costs associated with computer power, time and criminality, etc., the algorithm is considered safe. Furthermore, algorithm complexity confers intrinsic protection: if it takes a long time to crack the algorithm, such that data secrecy is no longer required, this is also safe (Schneier, 2015). Safety is also implicit if the volume of encrypted data with a key is less than the data required to breach the algorithm, and unconditional security is assumed if it is complex to recover plain text even when possessing substantive ciphertext (i.e. if the attacker can access scrambled data but cannot decipher it). In such circumstances a ciphertext-only attacker would have to try every conceivable key individually to analyze whether the resulting plain text is meaningful, which generally requires massive human attention in addition to computer systems, thus it is known as 'brute force' attack, beyond the capabilities and interest of most attackers (Goodrich & Tamassia, 2011). Cryptography is conventionally targeted to cryptosystems whose breach is computationally infeasible in the context of available resources. For instance, an encryption scheme where insufficient information exists in the generated ciphertext to uniquely determine the corresponding plain text, regardless of the volume of ciphertext accessed by the attacker or their computational power. The complexity of reversing underlying cryptography determines conditionally secure algorithms' security (e.g. the ease of factoring large primes), (Goodrich & Tamassia, 2011). An encryption scheme is said to be computationally secure if: the time required breaking the cipher exceeds the useful lifetime of the information. And if the cost of breaking the cipher exceeds the value of the encrypted information.

A comparison of cryptography algorithms' salient features is shown in Table 2.1, based on previous research (Premkumar & Shanthi, 2014)

Table 2-1 Commonly used cryptography algorithm features

Algorithm	Block Size (Bits)	Key Size (Bits)	Speed	Security
DES	64	56	Low	Less
3DES	128	112, 168	Low	Less
RC2	64	8-128	Fast	High
RC6	128	128,192	Fast	Secure
AES	128	128,192, 256	Fast	More secure
Blowfish	64	32-448	Fast	More Secure

2.4.1.2 Attacks

Attacks are essentially crypto analysis attempts by unauthorized users. There are seven main types of attack, as explained below, whereby it is assumed attackers (i.e. crypto analyzers) have comprehensive knowledge of algorithm used for encryption.

1. Ciphertext only attacks: the analyst only recognizes ciphertext to be decoded and attempts to discover the encryption key or to decrypt particular parts of ciphertext by analyzing several encrypted messages (Daernen & Rijrnen, 2002). The attacker seeks to revert to the plain text for particular messages or to identify the encryption key used in order to read data outputs.

2. Known Plaintext attack: this type of attack greatly facilitates the work of the attacker compared to ciphertext-only attack. When the attacker has specimens of cipher and plain text, they can attempt to identify the algorithms used to encrypt data, and thus obtain the key (Singh, February 2012).

3. Chosen Plaintext Attack (CPA): in CPA the attacker has chosen parts of plain text, as in differential cryptoanalysis, where the attacker is identified as being at the encryption site (Goodrich & Tamassia, 2011).

4. Chosen Ciphertext Attack (CCA): in CCA the attacker possesses chosen ciphertext and the matching plain text, facilitating decryption from the private key (Goodrich & Tamassia, 2011). Having obtained the chosen messages, it has access only to an encryption machine.

5. Chosen text: the attacker has the encipher algorithm, the ciphertext to be decrypted, chosen plain text messages, corresponding ciphertexts, fabricated ciphertext, and corresponding decrypted plain texts, developed by the private key.

6. Brute force attack: this is a dedicated, laborious and costly trial-and-error method, as described previously, whereby the attacker seeks to obtain the encryption key by running one-by-one encryption iterations using automated software generating vast numbers of guesses, usually entailing super machines that render such attacks unprofitable in most cases (Goodrich & Tamassia, 2011).

7. Man in the Middle attack: in this attack, communication between two parties is covertly intercepted and relayed (potentially in altered form) between the users, which can enable attackers to trick senders and intercept the secret key (Wikipedia).

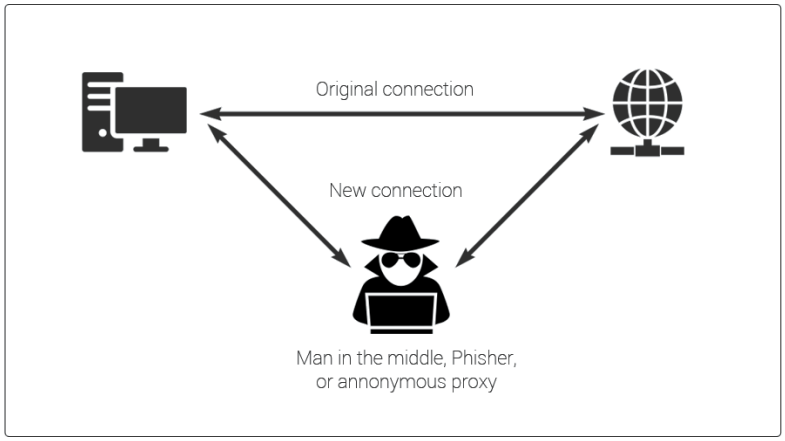


Fig. 2-17 Man in the Middle attack (Comado, n.d.)

The frequency analyst (differential attacks) endeavors to extract plain text by looking at the frequency of characters in ciphertext, based on the probability and prevalence of characters in messages (Memon, Rozan, Uddin, & Abubakar, 2014). For instance, 'E' is the most prolific letter used in the English language (Figure 2.11) (Oxford, 2017), thus if 'D' is the most frequent letter in a ciphertext this is likely to be the scrambled version of 'E', which enables the attacker to subsequently identify other letters and thus crack the code. Table (2-2) below show the frequency for each letter in the English language:

Table 2-2 Frequency of alphabet letters in English

Letter	Frequency	Letter	Frequency
e	0.12702	**f**	0.02228
t	0.09056	**g**	0.02015
a	0.08167	**y**	0.01974
o	0.07507	**p**	0.01929
i	0.06966	**b**	0.01492
n	0.06749	**v**	0.00978
s	0.06327	**f**	0.02228
h	0.06094	**g**	0.02015

r	0.05987	**y**	0.01974
d	0.04253	**p**	0.01929
l	0.04025	**b**	0.01492
c	0.02782	**v**	0.00978
u	0.02758	**k**	0.00772
m	0.02406	**j**	0.00153
w	0.02360	**x**	0.00150

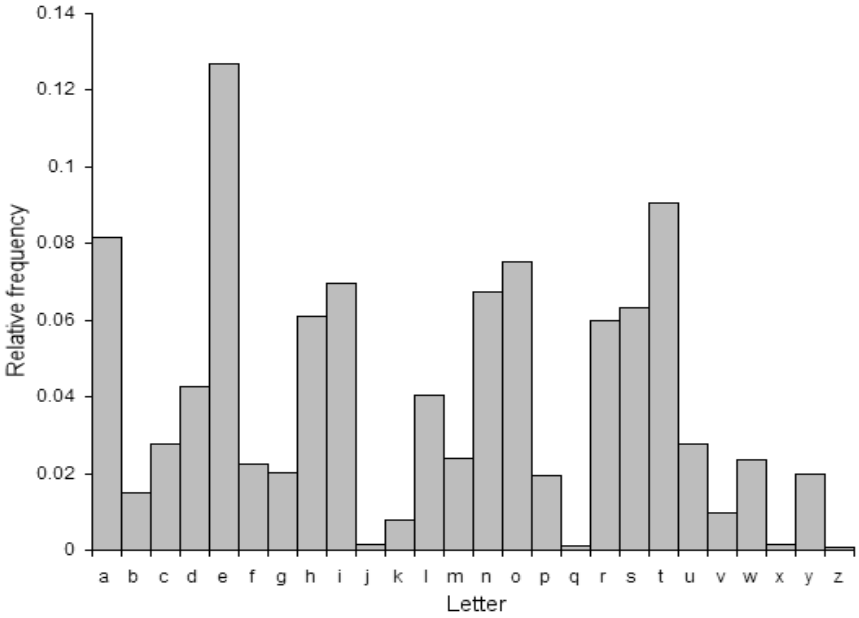


Fig. 2-18 Frequency of alphabet letters in English

2.4.2 Encryption in Wireless Devices

As explained previously, the most widely used key encryption algorithms in wireless networks are symmetric, because of asymmetric keys depend on mathematical functions that entail high computation loads and low efficiency for small wireless devices (Salama & Hadhoud, 2010)**.** Among symmetric key encryption algorithms amenable to wireless networks, RC4 and AES are highly efficient. Developed in 1987 by Ron Rivest, RC4 is a stream cipher used in numerous applications and wireless networks, including IEEE 802.11 WEP. It is efficient and fast, and highly effective into the early 2000s (Prasithsangaree & Krishnamurthy, 2003). It uses Wired Equivalent Privacy (WEP) protocol to provide standardized security services in wireless local area networks (WLANs). However, it now has numerous security weaknesses (Popov, 2015), (Fluhrer & Mantin, 2001). RC4 deficiencies in WEP protocol led to developing a new security standard in WLANs (IEEE 802.11i), AES.

Encryption by AES is flexible, fast, and compatible with many platforms, with particular utility in smart cards and small devices, for functions including routing information, request-response packets, and IP address encryption (between wireless nodes). AES was extensively tested for potential loopholes in security prior to its release, and it is considered highly secure for the modern wireless environment (NIST, 2004).

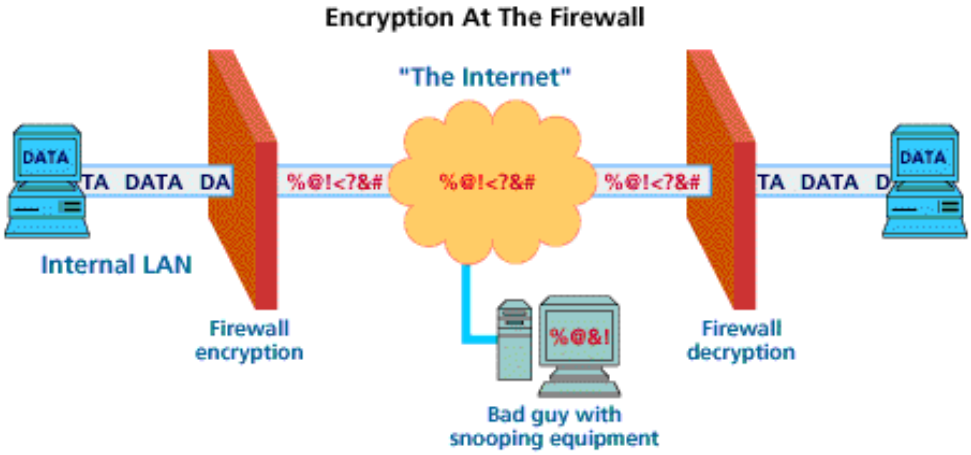


Fig. 2-19 IPsec (purevpn, 2017)

2.4.3 Energy Consumption in Encryption and limitation

Data encryption algorithms are applied in contexts where speed and communication efficiency is essential in addition to high-level security, including in online transactions, banking, and e-commerce (Nadeem & Javed, 2005). Wireless devices have limited resources in terms of electrical power (i.e. battery), memory, and processing power. While online and encryption technologies have increased remarkably since the early 2000s, there has been relatively slower progress in battery technology, a design problem known as the "battery gap". Mobile devices such as smartphone are most commonly used to access wireless networks, and they have niggardly power budgets due to the array of consumer-oriented applications they support, making them increasingly incompatible with modern potential security solutions, due to their computationally intensive encryption/decryption algorithms (Jiehong & Detchenkov, 2016).

For instance, wireless internet access itself is one of the major consumers of battery power, making access to wireless sensor networks problematic without access to a constant power supply and recharging facilities. Consequently, the higher power consumption required for the number of rounds in AES encryption is anathema to mobile device capabilities. Figure (2-21) illustrates the relationship between power requirement and security in relation to

encryption rounds. Clearly, it is essential to find optimal security relative to optimal performance and user satisfaction by developing the most power-efficient encryption algorithms for wireless networks.

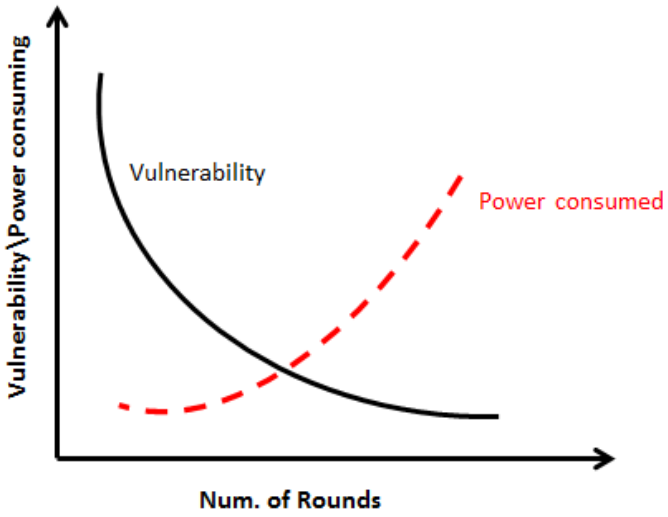


Fig. 2-20 Energy and Round Relation

Traditional cryptography failed to properly account for QoS requirements and device-specific needs such as power consumption, because prior to the 2000s the majority of internet/ wireless network access was achieved by desktop computers connected to mains electricity, and reduced speed was taken for granted as a necessary part of an already slow internet service (relative to modern fiber-optic broadband etc.). In the modern internet environment, users expect high-quality and fast service, regardless of the limitations of their own consumer devices (e.g. laptops and smartphones), thus researchers are focussing on increasing the number of computations used in network security with minimum latency times and power consumption (Chandramouli, et al., 2006).

One technique is to reduce transmission cost and complexity by using less data for permutation, by using only global encoding vectors (GEVs), and not whole message symbols. This method uses algorithms for random permutation confusion key calculation and key generation, achieving improved encryption time, energy consumption and throughput (Khan, et al., 2017) P Coding is another lightweight encryption coding scheme using permutation (Zhang & Lin, 2014), introduced a permutation coding scheme, called P Coding, which is a lightweight encryption.

Some hardware devices can reduce power consumption, but pipelined and loop-unrolled hardware usually requires a larger area and (paradoxically) high power consumption, and such architecture cannot achieve optimal feedback operational modes (Hamalainen, et al.,

2006) Common algorithms such as AES and DES have massive requirements for memory, that cannot be implemented in embedded systems (Bansod, et al., 2015) Lightweight encryption algorithms would have greater efficiency in wireless networks, such as PRESENT, but they have demonstrable vulnerabilities to attack (Lee, 2014), including full-round attacks using biclique cryptanalysis (Faghihi , et al., 2015).

A significant empirical study by Ramesh and Umarani (2012) compared the performance of different key sizes of AES (128, 192 and 256 bit) and UR5 (128, 192 and 256 bit). In terms of AES, they found that greater key size is associated with a change in time and battery consumption. Increasing from 128 to 192 and 256-bit keys increase power by 9% and 17%, respectively (Figure 2.21). (Jiehong & Detchenkov, 2016; RAMESH & UMARANI, 2012)

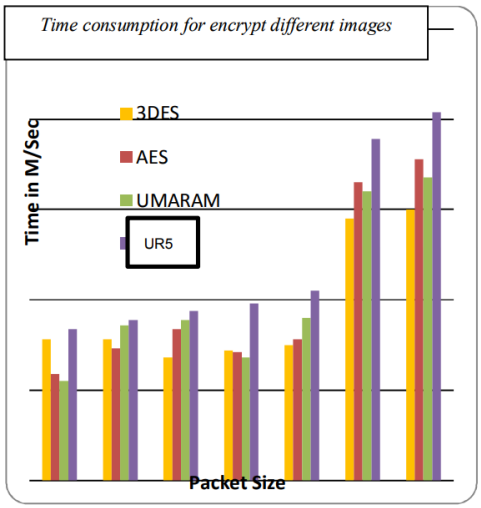


Fig. 2-21 (RAMESH & UMARANI, 2012)

Existing solutions proposed by researchers to reduce network costs (Scarfone, et al., Novmber 2007) are deficient in security terms by failing to encrypt whole messages, while AES encryption to modify the transmission protocol without modifying the actual AES algorithm is not cost effective (Zhang, et al., 2014). Consequently, this research aims to develop a current AES algorithm to provide high security with low power consumption in order to make a cost-effective solution.

2.4.4 Advanced Encryption Standard (AES)

The development in the arena of IT and the increased requirement of communication through various networks results in less demand for DES, as it is no longer suitable to meet the required demands. This has led to the introduction of Advanced Encryption Standard (AES), a new standardization which was introduced in 1997 by NIST. AES is a round-based

symmetric block cipher (Trichina, et al., 2005) which is required to exchange data over different networks. AES is also known as Rijndael (Zhang, et al., 2014).

AES (Advanced Encryption Standard) is the best cryptography algorithm using in computing, to protect the confidentiality of data. There is no successful attacks against AES have been recognized since 2004. The United State Government announced in June 2003 that the block cipher AES (Advanced Encryption Standard) algorithm for 128-bit key length is categorized as SECRET, TOP SECRET level for information.

In Nov. 2001 the National Institute of Standard and Technology (NIST) state that "*The Advanced Encryption Standard (AES) specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information. Encryption converts data to an unintelligible form called ciphertext; decrypting the ciphertext converts the data back into its original form, called plaintext. The AES algorithm is capable of using cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits.*" (Dworkin, et al., November 2001) Encryption is a fundamental tool for the protection of sensitive information (Nagaraj, et al., 2015) the attacker could be also identifying user actions in a network application. According to (Conti, et al., 2016) people continuously carry wireless devices with them and use them for daily communication activities, including not only voice calls and SMS, but also emails and social network interactions. The adversary may analyse the encrypted traffic in the network. The authors advised that a set of countermeasures should be taken and it may require a kind of trade-off between power efficiency and the required privacy level.

The AES encryption algorithm is composed of four transformation functions; AddRoundKey, substitution byte (SubByte), MixColumns and shift rows (ShiftRows). This iterative algorithm contains different rounds which are dependent on key length. There were ten, twelve and fourteen rounds used for key sizes 128, 192 and 256 bits. In AES every plaintext encrypted block consists of 128 bits, known as State Matrix and is illustrated through a 4x4 bytes square matrix. The majority of these AES tasks are completed over a predetermined and fixed field (Scarfone, et al., 2007).

The aforementioned four standard transformations for AES algorithm are discussed in more detail below:

1. **AddRoundKey:** This is a simple function and the most basic form for users. It uses a simplistic bitwise XOR operation. Every 128-bit round key XOR with a 128-bit state

matrix (Stallings, 2017). Figure 2.23 illustrates the simplistic structure of the state matrix with the present key to produce the output.

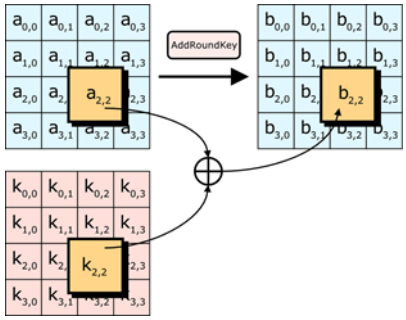


Fig. 2-22 Add Round Key transformation in AES.

For example the XOR of two Hex matrices:

$$\begin{pmatrix} 7d & 5c & ff & 76 \\ aa & 12 & 34 & 56 \\ df & aa & 11 & aa \\ a1 & b2 & 34 & aa \end{pmatrix} \oplus \begin{pmatrix} a1 & c4 & 6e & 00 \\ a6 & aa & 3a & ad \\ 43 & 3f & f6 & b8 \\ de & 9f & a4 & aa \end{pmatrix} = \begin{pmatrix} \boldsymbol{dc} & 98 & 91 & 76 \\ \boldsymbol{0c} & b8 & 0e & \boldsymbol{fb} \\ \boldsymbol{9c} & 95 & e7 & 12 \\ \boldsymbol{7f} & 2d & 90 & 00 \end{pmatrix}$$

When column 1 matrix 1 XOR column 1 matrix 2, then:

$7\text{d} \oplus a1 = dc$
$\text{aa} \oplus a6 = 0c$
$\text{df} \oplus 43 = 9c$
$\text{a1} \oplus de = 7f$ and so on.

Matrix 1 can be converted to binary form as:

01111101	01011100	11111111	01110110
10101010	00010010	00110100	01010110
11011111	10101010	00010001	10101010
10100001	10110010	00110100	10101010

2. **Sub-Byte:** Within this part of the process every byte within the state matrix is substituted with a SubByte through the use of 8-bit data from the AES S-Box. In the reverse procedure for SubByte, every byte within the matrix is substituted with relative inverse Sub Byte. Sub Byte processes result in changes which are not linear within the code (Abhijith, et al., 2013). There were two methods implemented to enable this procedure; multiplication with the irreducible polynomial and addition through an affine transformation (Stallings, 2012).

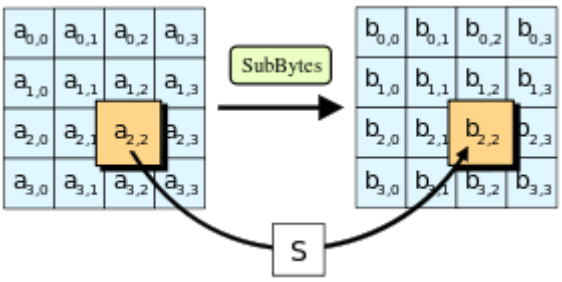


Fig. 2-23 Sub Bytes transformation in AES.

Table 2-3 The AES S-Boxes Table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

3. **Shift-Rows**: The transformation during the ShiftRows stage is a permutation function which enables cycle shifts for each row within the state matrix. This means that the top row of the state matrix remains the same, whereas the other rows are moved in relative cyclical patterns (Daemen & Rijmen, 2003). This pattern of change is imperative and ensures that the previous four columns create the next byte of the state matrix (Smith, 2003). This process is shown in Figure (2.25) demonstrates this operation

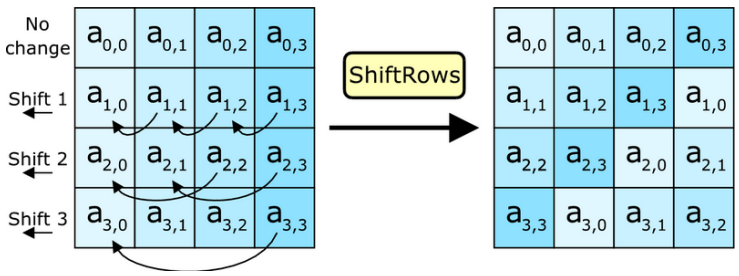


Fig. 2-24 ShiftRow Transformation in AES Algorithm

4. **The MixColumn** method of conversion is a significantly complex function compared to the other methods discussed within AES. It is illustrated in Fig 2.26 and the transformation occurs column by column within the state matrix. Every vertical column is addressed ‘as a four-term polynomial over GF(2^8) and it is multiplied by the constant polynomial value (Stallings, 2017).

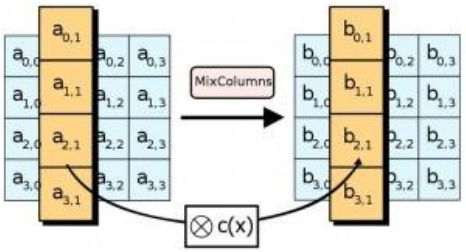


Fig. 2-25 The Mix Column transformation in AES

For example the multiply of two Hex matrices:

$$\begin{pmatrix} 03 & 02 & 01 & 00 \\ 02 & 03 & 02 & 01 \\ 01 & 02 & 03 & 02 \\ 00 & 01 & 02 & 03 \end{pmatrix} \bullet \begin{pmatrix} a1 & c4 & 6e & 00 \\ a6 & aa & 3a & ad \\ 43 & 3f & f6 & b8 \\ de & 9f & a4 & aa \end{pmatrix} = \begin{pmatrix} xx & yy & zz & gg \\ 0c & b8 & 0e & fb \\ 9c & 95 & e7 & 12 \\ 7f & 2d & 90 & 00 \end{pmatrix}$$

The result for the first row is as follows:

$$(03 \bullet a1) \oplus (02 \bullet a6) \oplus (01 \bullet 43) \oplus (00 \bullet de) = xx$$

$$(03 \bullet c4) \oplus (02 \bullet aa) \oplus (01 \bullet 3f) \oplus (00 \bullet 9f) = yy$$

$$(03 \bullet 6e) \oplus (02 \bullet 3a) \oplus (01 \bullet f6) \oplus (00 \bullet a4) = zz$$

$$(03 \bullet 00) \oplus (02 \bullet ad) \oplus (01 \bullet b8) \oplus (00 \bullet aa) = gg$$

and so on.

Matrix 1 can be converted to binary form as:

00000011	00000010	00000001	00000000
00000010	00000011	00000010	00000001
00000001	00000010	00010001	00000010
00000000	00000001	00000010	00000011

The polynomial representation: this is a method can be performed by multiply each bit by X. the advantage of using this method is to represent the numbers in GF. Fig. (2-27) illustrate the multiplication of two binary numbers.

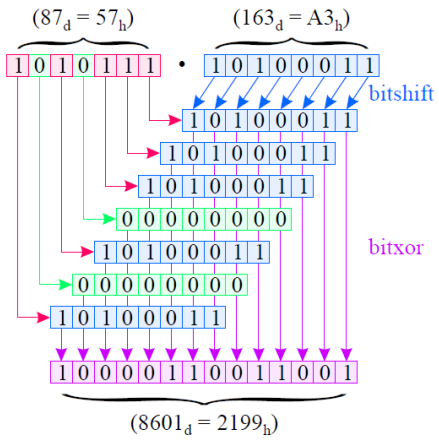


Fig. 2-26

Unfortunately, the resulting polynomial has a degree greater than 8 bit, can therefore not be expressed in one byte (i.e. it is not a GF(2^8) element) and has to be transformed back into the ”byte range” by the modulo division (Stallings, 2017).

2.4.5 AES Encryption and Decryption

The process for the encryption of the AES algorithm begins with the addition operation of input data and round subkey via an XOR operation. This is followed by a substitution-permutation network (SPN) consisting of the four transformation operations (Stallings, 2017). This encryption method is seen in Figure 2.28.

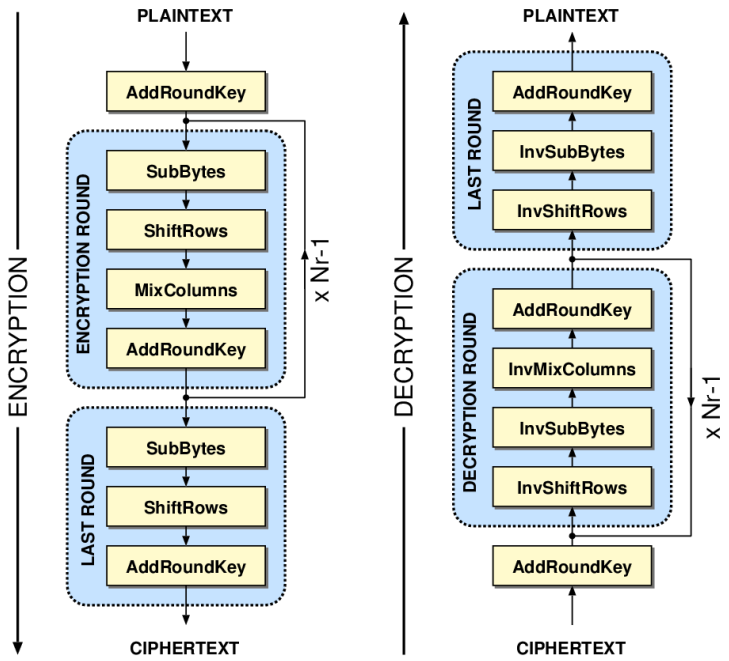


Fig. 2-27 AES algorithm

The AES decryption algorithm for each method processes similarly through inverse transformations of the encryption algorithm as described previously. For example, a replicated reverse transformation is used in the decryption process as previously seen used in the encryption process for AddRoundKey. The subsequently created state is identical to the present state exclusive XOR with the key, which means the inverse state is applied in both encryption and decryption.

Inverse-ShiftRow is the term used for the inverse of ShiftRows. This means that the initial row of the state matrix remains unchanged. However, rows two, three and four are moved by one, two and three bytes respectively to the right (Trichina, et al., 2005). This operation is illustrated in Fig 2.29.

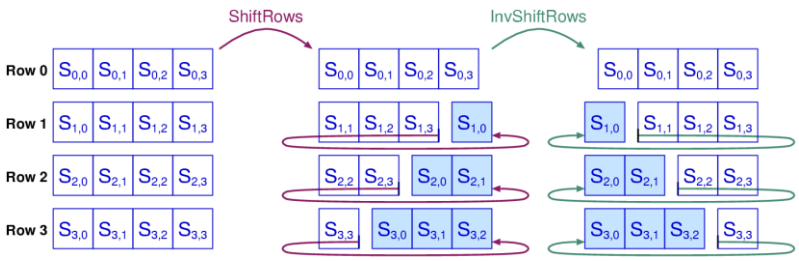


Fig. 2-28 Shift row

Inverse-SubByte inputs the Inverse S-Box data, which are illustrated in Table 2.4. The modification uses a one to one mapping process which is the antithesis of the SubBytes task used in the encryption procedure (Stallings, 2017).

Table 2-4

		y														
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

The Inverse-MixColumn illustrated in fig (2-28) also used the polynomial representation but with inverse constant polynomial (Veena, et al., 2016) :

$$\begin{pmatrix} 14 & 11 & 13 & 09 \\ 09 & 14 & 11 & 13 \\ 13 & 09 & 14 & 11 \\ 11 & 13 & 09 & 14 \end{pmatrix} \bullet \begin{pmatrix} s \\ s \\ s \\ s \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ g \end{pmatrix}$$

$$(14\bullet s) \oplus (11 \bullet s) \oplus (13 \bullet s) \oplus (09 \bullet s) = x$$

And so on,

The following fig (2-30) shows how the characters are represented in Hex system and relation to ASCII code for computer system and programmer functions. The first column contains a serial hexadecimal number of the position of the characters displayed in the other two columns.

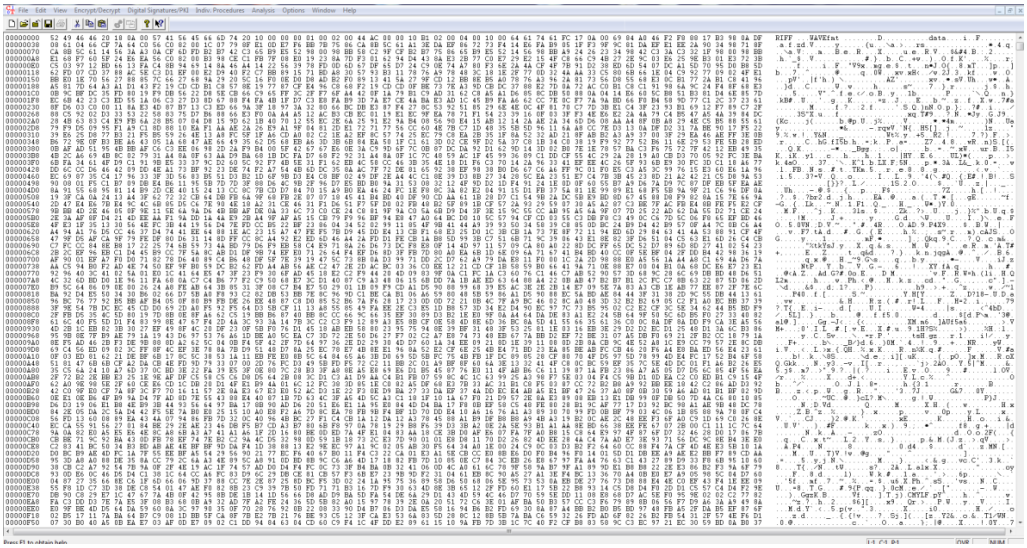


Fig. 2-29 File Representation

In the second column the characters are presented in hexadecimal form (see ASCII table). One character is represented by two characters one after the other (0, 1,..., 9, A, B,..., F).
In the third column, the displayable characters are shown in accordance with their ASCII code. Non-displayable characters are depicted by a dot.

As mentioned before, AES is in a good choice for wireless networks because it is symmetric encryption algorithm and more secure, however, there are some issues in the time and power consumption making it critical in an application such as wireless sensor networks. The figure below shows a comparison of some algorithms.

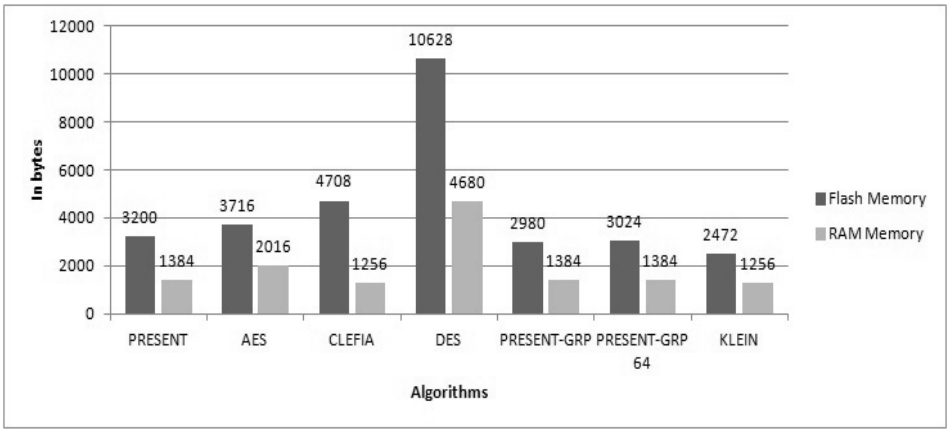


Fig. 2-30 (Bansod, et al., 2015)

Current research has focused on OpenMP directives and stream cipher in addition to AES block cipher encryption. Hosseinkhani and Javadi (2012) used cipher key for dynamic S-Box generation, with static S-Box increasing AES cipher cryptographic strength, generating novel S-Boxes when necessary merely by changing two bits of the cipher key. For almost all block cipher applications AES is an excellent solution, but it is deficient in high demand, constrained environments, including sensor networks and RFID tags. Furthermore, smartphone battery life is highly reduced by AES encryption (Masoud, Jannoud, & Ahmad, Sep 2015) , although according to some studies it provides the best security relative to power demand (Bogdanov, Knudsen, Leander, & Paar, 2007), particularly compared to 3DES (Ahmad, et al., 2016).

2.4.5.1 Key Expansion

Key schedule or expansion (according to the number of rounds) is a necessary prerequisite of the encryption process. For instance, a 128-bit (16-byte) key might be expanded into an array of 44 (32-bit) words; given a 128-bit key, this is arranged in an array of 4x4 bytes. As in the input block, the initial keyword is entered in the initial array column, and so on (Kak, 2018). The key array's four columns of words are thus expanded into a 44-word schedule, whereby every round uses four words from the key schedule. Figure 2.32 shows the original 128-bit key's four words being expanded into a 44-word key schedule. See appendix.

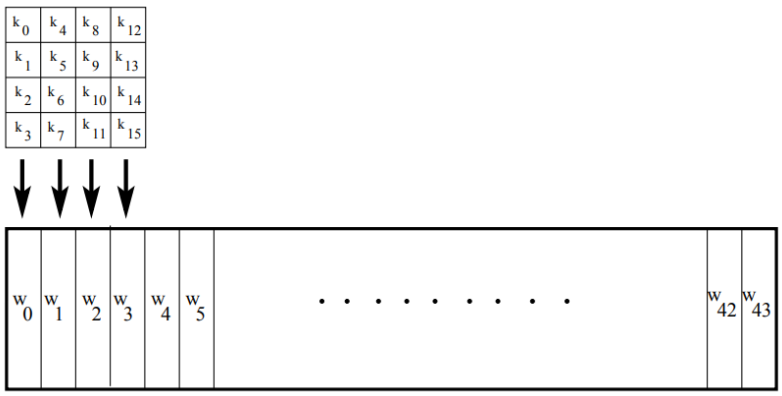
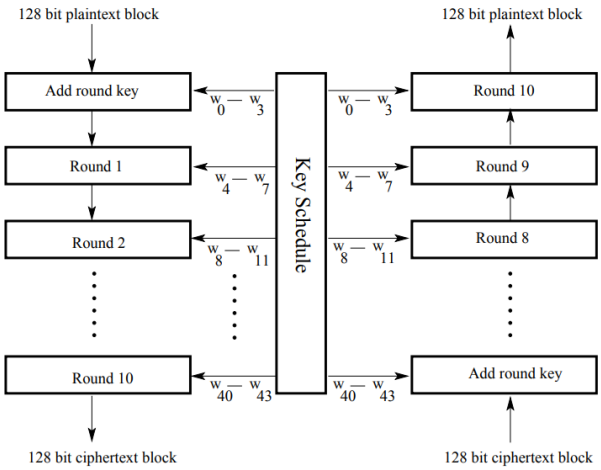


Fig. 2-31 (Kak, 2018)

It is designed to resist known attacks where knowing part key insufficient to find many more and to diffuse key bits into round keys.



AES Encryption AES Decryption

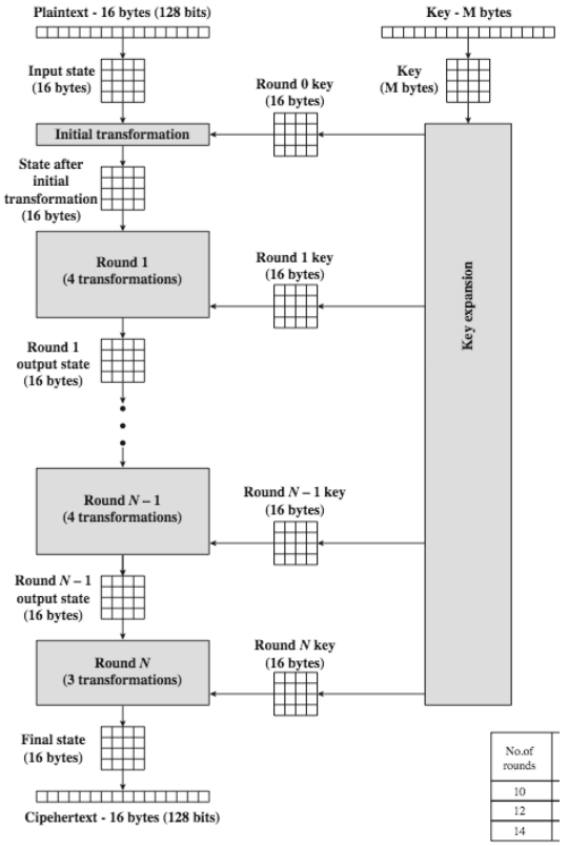


Fig. 2-32 Key encryption (Stallings, 2017)

2.5 Lightweight Cryptography Discussion

Many previous types of research have started to address security issues in wireless MANET, most of them could solve many problems in this network especially for real-time traffic which is crucial.

(Matesanz, et al., Nov. 2012) showed the most common threats to ad hoc networks and reviewed several proposals that attempt to minimize some of these threats, showing their protection ability and vulnerabilities in light of the threats that might arise.

Address Spoofing In this attack the malicious node intentionally select an allocated or a free IP address. When it takes the assigned IP of the victim, it will hijack its traffic. And in the other case, the node allocates the free IP address to itself to join in the network, trying to collect important information which helps to execute some attacks, like denial of service.

Address Space consumption in this threat the attacker can consume the address space by demanding a huge number of IP addresses. The attacker could request the assignment of IP addresses to fake nodes, so it could avert other nodes from being configured and arriving at the Mobile ad-hoc networks.

Address Conflict Threat In this attack the malicious node may assign an additional address to a client from possible addresses which already in use. So, it would produce many problems in the Mobile ad-hoc networks and lead to address conflict (Sandoval & Garc´ıa, 2012).

Denial of Service Threat where the attacker can, in an autoconfiguration process, act as a requester and send AREQ messages to several nodes simultaneously causing an overload of traffic.

Multiples Identities (Sybil) This kind of attack is possible when a repudiation system is to state the legality of nodes in the network. The attacker could cause the isolation of valid nodes by making messages to blacklist those nodes. Blackmail is a type of Sybil attack.

Layer	Attacks
Application Layer	data corruption, viruses, and worms
Transport Layer	TCP/UDP SYN flood
Network Layer	hello flood, blackhole
Data Link Layer	monitoring, traffic analysis
Physical Layer	eavesdropping, active interference

Another work by (Apietro , et al., 2014) survey emerging and established wireless ad-hoc technologies and they highlight their security/privacy features and deficiencies. They also identified open research issues and technology challenges for each surveyed technology.

(Binod & Hyuk, 2010) took the advantages of multipath which are useful for reliable MANET, and they proposed a basis for sending secure real-time streaming by using multipath mobile ad hoc network. It can offer security for wireless ad hoc routing and multimedia transfer. They considered the digital signature and encryption technique. Their results show a good performance, also they suggest in future work the using of multicast which is important in many video applications.

Furthermore, (Emmanouis & Christos, 2012) suggested a secure model for autonomous networks like mobile ad hoc networks to create real-time communication in disaster rescue operations. Secure P2PSIP, intrusion detection, and secure routing methods are a vital part of any secure model. The aforementioned concerns in the realm of emergency ad hoc networks have been discussed. They presented two additions of the IETF drafts and analyzed in relations of security power and scalability. The design of these extensions helps to meet the requirements of a mission-critical MANET where rescuers should create links to communicate with each other by using lightweight devices such as PDAs. The situation of adaptive routing protocol and its security addition by using IPsec have been also discussed.

In spite of these researches could address some security issues for real-time traffic in MANET. However, may could not address the cryptographic issues in this network.

(Sehgal, et al., 2011) in their recent research paper proposed an architecture, which states that self-organized distributed security could be provided by five-layer security model,

authenticated and security aware routing. After they have carried out the simulation, they found it provide security with negligible overhead.
A layered architecture can provide such advantages as modularity, simplicity, flexibility, and standardization of protocols. Five-layer security architecture for MANETs: Trust Relationship Security Layer, Node-To-Node Security Layer, Routing Level Security Layer, Network Level Security Layer and Application Level Layer. They expect this security architecture can be used as a framework when designing system security for ad hoc networks.

In another side, many previous types of research have been conducted to address cryptographic issues related to real-time traffic. They could develop some algorithms to deal with real-time limitation issues, like delays.
(Hu, et al., 2009) propose a relay encryption scheme based on a threshold secret sharing algorithm, to enhance the data confidentiality, reliability, and integrity in MANET. The basic idea is to distribute the whole encryption task to all nodes along the route, every node completes part of the encryption task instead of the single source node encrypting the whole data. They found that the encryption scheme can effectively alleviate the burden of the source node and reduce the transmission delay, besides, it improves the reliability of MANET
Also, (Mohammed & Rohiem, 2009), have introduced a Variable Mapping S-box technique with AES (VMS-AES) for voice over Internet Protocol VoIP. By using a key to generate a parameter that used to shift (remapping) the substitution of S-box to another location randomly depend on the initial key and the derived sub keys data. The results of this work could maximize the nonlinearity to provide more resistance against linear attacks and to solve the fixed structure of the substitution function. Also, Increase the overall security. It has also shown that 7 rounds of encryption consume less power and execution time without affecting the security level. However, their findings have been criticized long time ago, by (Ferguson, et al., 2001) and (Daernen & Rijrnen, 2002) who explained the possibility of saturation attacks on six and seven rounds of AES. They state that seven-round attack can be mounted by adding one round before the distinguisher and two rounds after it. For these reasons, their work was still unsuitable for the security requirements of wireless networks.

Another study has been published by (Rahma & Yaco , 2012) who proposed asymmetric dual key Dynamic block algorithm (SDD) for digital video in the partial encryption technology. This algorithm meets the requirements of real-time with a high level of complexity with considerable speed. The proposed encryption algorithm SDD achieves better results of the

time faster than AES by 13 times factor for encryption and 9 times of the decryption. This work has a high resistance to brute-force attack because it used two keys in the encryption process; however, it encrypts only selected parts of the data, which consider the risk of a breach in security. Also, In spite of the new features which have been achieved in their work, there is still a lack of security analysis in this work to prove its strength

(Das , et al., 2013) devised a new algorithm to generate random S-Box and its inverse S-Box based on using different irreducible polynomial in the finite field GF(2^8), while only fixed polynomial in AES standard. The irreducible polynomial in the AES standard is $m(x) = x^8 + x^4 + x^3 + x+1$, this polynomial is used to find the multiplicative inverse which is well known to any attacker. To overcome this problem in the proposed algorithm, the different irreducible polynomials are used every time in the finite field of GF (2^8) and send this to the receiver joint with the secret key to raise the security of the cipher operations. They tried to address security concern, and their results kept the execution time and energy the same, and this may not suit the new wireless environment.

In the same field, (Lambić & Živković, 2013) applied an alternative S-box generation method of forming compositions of permutations from some fixed sets. After choosing these sets, output S-boxes are obtained by making various compositions of the starting S-boxes. The sequence of the used indices of starting S-boxes is key-controlled.

(Barnes , et al., 2012) took a sequential program that implements the AES algorithm and converts the same to run on multicore architectures with minimum effort. Two different parallel programs have been implemented, one with the fork system call in Linux and the other with the PTHREADS, the POSIX standard for threads. Similarly, (Navalgund, et al., December 2013) proposed an optimized parallel AES algorithm which was implemented on shared memory architecture systems. The proposed algorithm uses two techniques for parallelization. The first approach uses data level parallelization, while the second one uses control level for parallel. The proposed algorithm is executed on the multi-core system and the program is written in C language, OpenMP standard is embedded into the program to get full parallelization benefits. The results show very attractive performance-effort ratios by OpenMP. Also, (Nagendra & Sekhar, 2014), implemented the AES algorithm using OpenMP to parallelize this algorithm. The OpenMP standard directives are embedded to an existing program to reproduce a new parallel version of that program. The parallel AES is implemented on a dual-core (Intel Core 2 Duo) system.

The study by (Ali, et al., 2014) proposed a new encryption/decryption algorithm based on the AES scheme. They suggest some modification on two AES functions, subyte, and mixcol, to increase the security of the encryption strength and keep the execution time at the same level to maintain the QoS of real-time traffic. Their results achieved a significant level of complexity in the new algorithm (8!*4!*256!) and this made it more complicated to be breached by the parasitical. However, keeping the execution time at the same level, as in AES, is not enough for current wireless environment requirements.

Furthermore, a new proposal by (Masoumi & Rezayati, February 2015), proposed a novel method for implementation of the advanced encryption standard (AES) algorithm, which offers an improved strength against differential electromagnetic and Energy analysis with minimal additional overhead. Their results showed significant protection for some microprocessor-based security tokens with limited resources such as smart cards. In this research, the authors explained the power analysis attacks and how it is used to compromise the security of cryptographic hardware, such as power side-channel and EM attack. They also mentioned a vital point on how the output of first SubBytes is usually attacked in practice since that is the only function in AES in which data and cipher key enter a direct operation *which what is being tried to address in this research*. The authors still concern about a lot of questions remain open, regarding the hardware security of microprocessor-based tokens and hardware security modules which is still a challenge, costly and must be done with care.

(Bansod, et al., 2015), designed a new lightweight compact encryption system based on bit permutation instruction group operation (GRP) they proposed a new hybrid system that offers more compact results in terms of memory space and gate equivalents in embedded security. The authors frequently stated that the standard algorithm such as DES & AES have vast memory requirement and would not be possible to be implemented for embedded system scheme.

Finally, the research conducted by (Msolli, et al., July 2016) suggested a 5 rounds AES encryption algorithm for multimedia and real-time applications in a wireless sensor network. Their aim was to reduce the execution time of the encryption process to be suitable for WSN nodes requirements. The results showed a good execution time has been reduced. Also, the security analysis of histogram showed good encryption strength and randomness. However, the authors didn't know from the literature that a last successful attack on 7 rounds has been

claimed by (Bahrak & Aref, July 2008) So this work still critical in term of security and cryptanalysis.

So, the aforementioned work is still has some gaps in addressing the full lightweight cryptography solution in wireless environment. And there is no actual considerable balance among the security and quality parameters such as latency energy and complexity strength of cryptosystems. The following section is critically addressing these gaps in the previous studies.

2.5.1 Summary & Conclusion toward Thesis Knowledge gap

From this chapter, we can summarize that Wireless networks are the most important part of IoT, and their security is crucial. The encryption is the main principle of security because it keeps the confidently of the information, and the best encryption algorithm is AES algorithm because of its strength. However, this algorithm needs more research to meet the QoS requirements of the new wireless devices such as the energy of these devices and the time (latency) for voice traffic. The security system designed for IOT should be able to detect and prevent both internal and external attacks. And as mentioned, not all linked devices have enough computational processing energy. That means tasks like encrypting data are going to be impossible and any type of security must be lightweight. The privacy of the information on IOT needs a reliable security system that prevents unauthorized access to private data on the network. Cryptographic mechanisms must be smaller and faster but with little or no reduction in security level.

Real-time applications and voice are very important in wireless networks because huge traffic of these applications will use the wireless networks in IoT. So secure these applications and keeping their quality is a big challenge. As explained, there are many types of research have been done to address these issues such as, (Mohammed & Rohiem, 2009; Rahma & Yaco , 2012; Lambić & Živković, 2013) however, they still have some gaps in their research. Also, Unfortunately, some of the researches (Ali, et al., 2014) and (Das , et al., 2013) focused on security concern and they raised the complexity of encryption algorithms to make it more complicated and high resistance against cryptanalysis but they didn't give further attention to QoS requirements and nodes limitations, and this made them not suitable for wireless devices, because of the sensitive requirements for wireless networks, like delays, throughput, and power consumption. In spite of the work of (Bansod, et al., 2015) and (Msolli, et al., July 2016) who have addressed the QoS metrics, but their work has a lot of concerns regarding the

security strength. Specification of a good algorithm has to be lightweight and secure to be suitable for the wireless environment. The other gap in their work is that there is still a lack of convincing security analysis to prove the security strength of the proposed schemes.

In our point of view, the best solution is to propose a new approach to deal with these requirements by modifying some functions in the current cryptography algorithms; this then becomes lightweight and suitable for V-over-WMANET and helps to get a good tradeoff between the security and QoS metrics. The problems in the voice delay and in the power of devices and in the security level of algorithm which can be solved by this thesis because previous work couldn’t solve it together. The following chapter is going to explain the methodology of the proposed approach and its elements and tools, to achieve the main objective of this research.

3 Chapter Three: Research Methodology & Proposed Framework

3.1 Introduction

The literature review of the security and cryptography in wireless networks presented in the previous chapter obviously identified a research gap: encryption techniques are not suited for wireless traffic, costing more resources; also, the weakness of AES is that it works with a single key. In particular, this research addresses the lack of knowledge on how to secure the information with cost-effective ways. All researches are based on basic assumptions about what establishes valid research and which research approaches are applicable for the development of the knowledge in a specific field. The research is a systematic work undertaken to increase the standard of knowledge, and the use of this standard to invent new claims (Manual, 2015). It is used to create or confirm facts, confirm the results of previous work, support hypotheses, resolve new or existing problems, or develop new models. A research mission may also be an extension of past work in a specific area. The approach in which someone chooses to measure something is called the methodology.

3.2 Research Quantitative Methodology

The key characteristic of any research is measurement. This is based upon a fact that things can be measured and measured reliably. However, some people disagree with this fact, arguing that certain features of human behaviour and experience cannot be objectively measured. So, they adopt a *qualitative* approach to research. This type of approach does not narrowly focus on a specific question but consider the theoretic logical model in an inquisitive, open-ended settle in proves as they adopt a perspective (Neuman, 2011). The other method that has been used to conduct research is Quantitative approach. Usually, Quantitative methodologies contain experiments, observation, and structured interviews. Quantitative researchers usually begin with a general area of study or personal interest or issue of professional. "*Researchers must narrow it down to, or focus on, a specific research question that can be addressed in the study. Often this requires a careful review of the research literature and developing hypotheses that frequently come from social theory*" (Neuman, 2011). The Strengths of the Quantitative Research Methodology Approach has been addressed in (Choy, 2014), the author states that this methodology has the reliability by

critical analysis. Also, enabled numerical data for groups and extents of agree or disagree from respondents. This research will focus on quantitative approaches.

3.2.1 Justification of the Research Method

This research adopted an investigatory strategy, to employ observational investigation by using quantitative methodology. The experimental approach was planned in accordance with scientific technique. The research objectives were defined in order to collect primary evidence from literature, in addition, to obtaining empirical proof over experimentation, in order to justify the design elements for a TKE algorithm. The prototype was constructed from these design elements and tested to compare their performance against the standard technique and the similar state of the art researches carried out by other authors. The analysis of empirical data obtained from tests allowed the research questions to be answered. A quantitative research method helped in this study to interpret the statistical security analysis by collecting the numerical data. It also gave a good comparative study with current algorithms.

3.3 Rationale for Research Approach

There are various evidence could be clearly seen throughout worldwide media of the threat of cyber-attacks on people privacy and security. The literature review provided considerable evidence that clearly showed the security requirements for next-generation wireless networks and their limitations (Chapter 2). In addition to the security protection, the implementation cost like execution time and energy consumption should be maintained as well, particularly for multimedia and real-time data. This is the big challenge which should be addressed in the research when how the balance should be achieved between the security and QoS metrics. The literature has also provided evidence about encryption importance in the security field (Objective#1). The Encryption of the data represents the backbone of security and privacy. The main principle of security is confidentiality, which can be achieved by encryption (Merkow & Breithaupt, 2014). Encryption algorithms have been used in many applications and protocols. They have been used to encrypt data transferred into the network and could be used to encrypt the IP addresses as well. The routing information and request-response packets between wireless nodes could also be encrypted. These features have made the encryption very important in cyber security. However, the encryption processes have many issues; according to (Jiehong & Detchenkov, 2016) encryption algorithms consume a significant amount of computing resources such as CPU time and battery power in mobile

devices. For instance, AES encryption process does the substitution processes to achieve the "*Confusion*" and it is doing repeated iterations to achieve the "*Diffusion*" in the cipher; also, it helps to hide the relationship between the key and text (Diffusion). But, this cost much more execution time and energy consuming which is limited in wireless devices (Section 2.9.3). AES is very secure because it used substitution, permutation, mixing, and keys, in addition to many rounds of iteration. These operations offer the confusion and diffusion needed to protect any cipher from cryptanalysis attempt. So, any proposed cryptosystem based on AES features will be efficient and secure. In this research, our proposed crypto-schemes are based on AES features because of its strength. The proposed algorithm TKE suggests five functions to implement the encryption process.

Objective#2, 3 has been built upon the claim by (Abhiram, et al., 2015) that Static S-Boxes are implemented using lookup tables which will never vary with the input text or input key. This technique makes reverse engineering very simple for the purpose of cryptanalysis. The sub-byte function and S-Boxes are used in cryptography in order to provide non-linearity in the design of cryptographic primitives such as block ciphers and hash functions (Alsalami, et al., 2016), "*the operation of an S-box cannot be encoded in a linear equation*" (Matsui, 1994). Similar to AES, S-Box creation is a pre-encryption process (Dworkin, et al., November 2001), so creating many S-Boxes will lead to increasing the algorithm complexity without affecting the execution time and power consumption of the encryption process; because it is pre-encrypt process and these S-boxes should be generated before the encryption starts as explained in (Chapter 5).

Review of the state of the art has shown that many encryption algorithms exist, with variable features. The existing works focusing on security concern while neglecting the wireless and voice limitation. Also, (Masoud, et al., Sep 2015) believe that utilizing the AES algorithm with their complexities in smartphone for social applications is not convenient. They state that a new encryption algorithm with less complex instructions should be introduced. So, research objective #4 has been built upon suggestions identified in the following literature. According to (Ali, et al., 2014), Mixcolumn function is a more expensive operation in AES, so the modification of this function will reduce their execution cost. In addition to the claim by (Daernen & Rijrnen, 2002) which applied to Rijndael AES, state that the saturation attacks is faster than an exhaustive key search for reduced-round versions of up to six rounds. Furthermore, this objective, built to address the round reduction proposed by (Mohammed & Rohiem, 2009) which is vulnerable to differential cryptanalysis attack. For

this reason, using 9 rounds will create TKE more resistance to cryptanalysis. It makes a good tradeoff between the complexity and Power consumption, compared with standard AES, without affecting the security and complexity of the new algorithm (Chapter 6).

Research objective #5 was built upon claim by (Abhiram, et al., 2015) that weakness of AES is that it works with a single key. Also, the recommendation of NIST about the key importance and key management (Scarfone, et al., Novmber 2007), gave further attention on key security because the confidentiality of the key determines the security of the algorithm. For instance, a man in the middle attack can fraudulently capture the cryptography key and use it to reveal the encrypted data. Therefore, a third key has been added to the proposed algorithm in order to increase the security level for it, this key is XOR with the output cipher-text in the last round only. The third key length is 16-byte, thus the key space for it is 2^{128}. Empirical data evidence was collected to demonstrate that newly proposed algorithm TKE is faster than AES and consumed less Energy.

In relation to the security analysis and evaluation, the effort previously spent on evaluating AES security strength has been very limited. (Preneel & Rijmen, 2000) said that the majority of researches at AES dealt with performance evaluation rather than with security evaluation. So, the security analysis has been deeply conducted in this study to prove the strength of the proposed schemes.

The security of the packet transfer using NETFPGA is one that has not yet been implemented on a more secure scale due to hardware limitation as many of the advanced algorithms such as AES or DES cannot be used due to their high complexity and excessive use of resources (Chouhan, 2016). Power analysis attacks are non-invasive and use power consumption profile to compromise the security of cryptographic hardware (Muresan, 2012). Although hardware implementations generally offer higher throughput and better energy efficiency than software designs, they are difficult to upgrade and adapt for future possible protocol changes (Baas & Liu, 2013). Moreover, the designs are very time consuming and costly. So, software implementation is more suitable in these applications.

3.4 Research Design

The approach to researching the problem of voice encryption in wireless devices is conducted in three phases. **Phase 1** has three stages: stage 1 was a review of network security and state of the art in encryption/decryption techniques. Stage 2 was an empirical statistical test of traffic in the wireless network. Stage 3 an experimentally test of AES encryption on voice; (phase 1 represents ch.2 and ch.4). **Phase 2** was the construction, justified through empirical statistical evidence, of a TKE algorithm, which was performance tested against different file sizes. It contains of three stages: stage 1 Multi S-box generation for security increment, stage 2 for time and power reduction, and stage 3 for triple key usage and 30% power saving, (phase 2 represents ch.5, 5, and 7). **Phase 3** was the Validation & evaluation of component elements for the construction of a modular framework architecture that allows TKE to be used for encrypts the voice over a wireless environment, in addition to the conclusion and future work. Please see fig. 3-2 for more clarification.

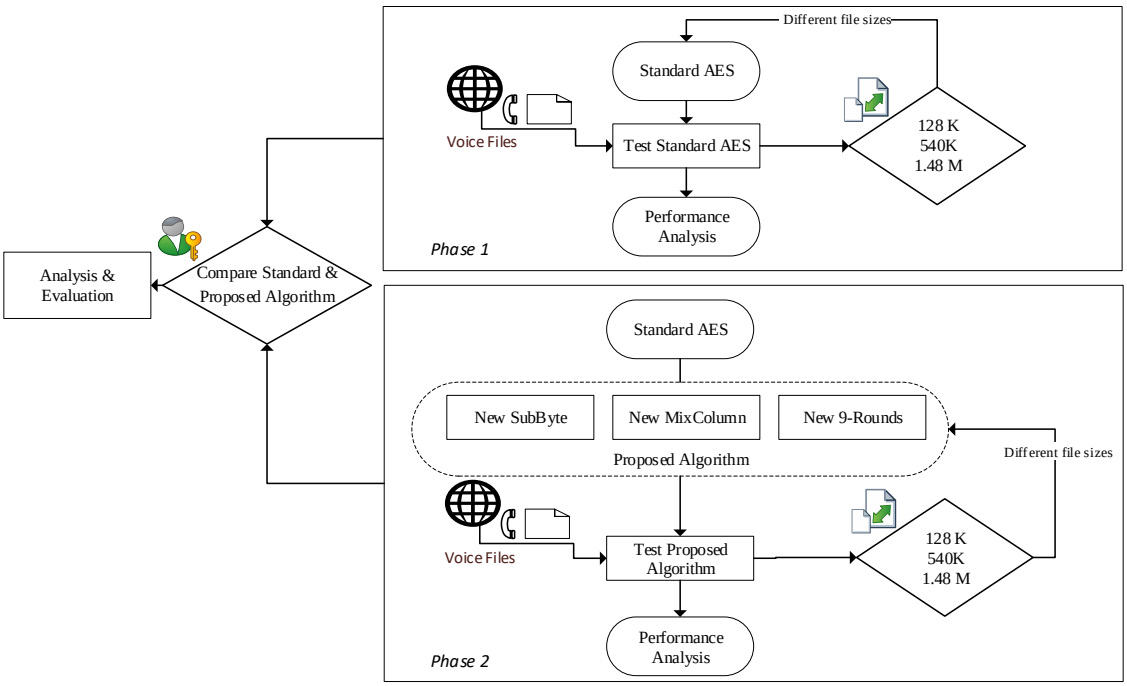


Fig. 3-1 Validation & Evaluation Method

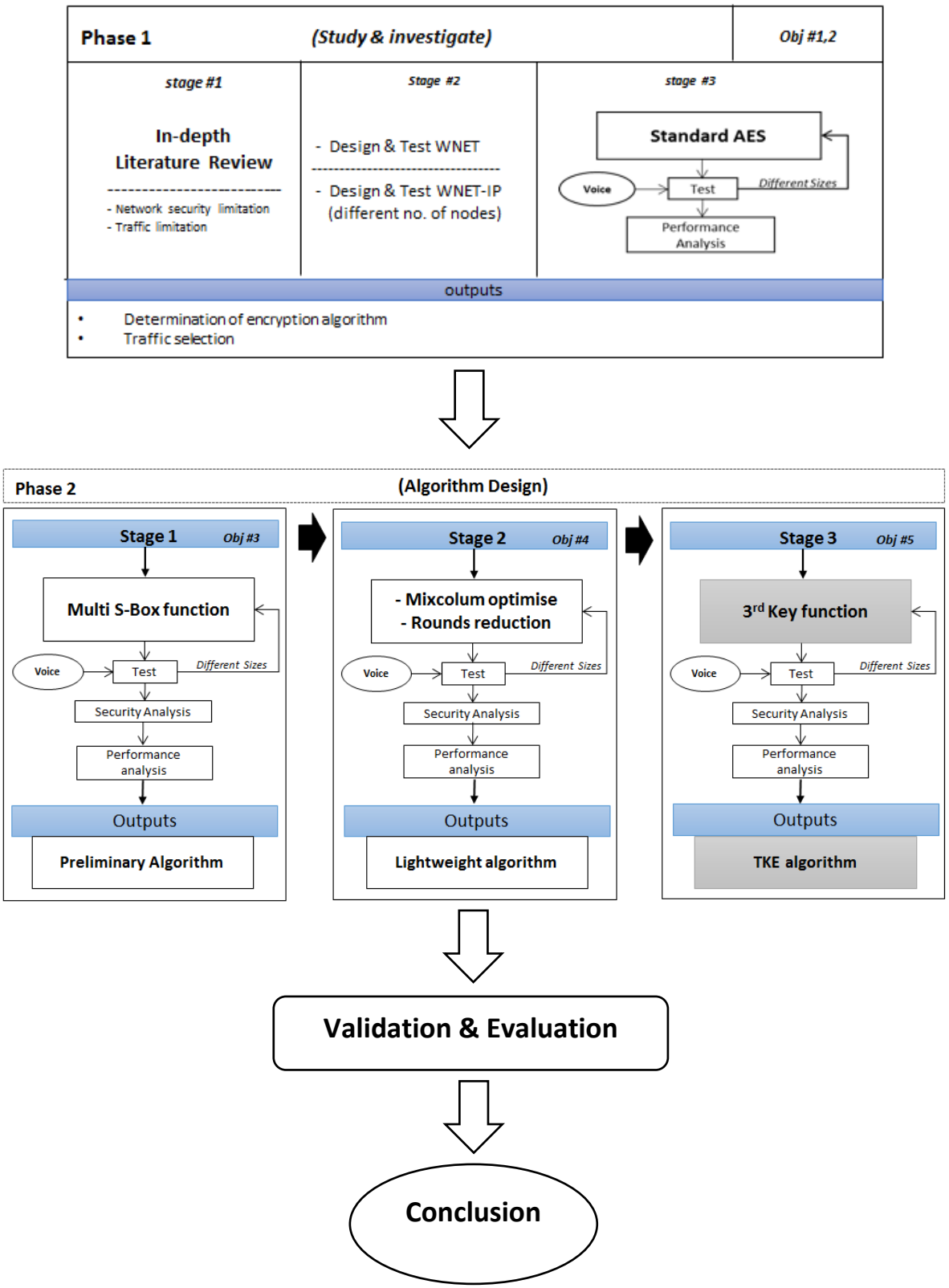


Fig. 3-2 Research Design

3.5 Implementation

This section explains the implementation of the research experiments. Usually, in the network environment, the encryption occurs in the nodes, so the focus will be on the encryption process in the nodes and the effect it has in terms of delay and power consumption. The implementation of the experiments and the parameters chosen according to (Zhang & Lin, 2014)

3.5.1 Proposed Framework

3.5.1.1 Data Investigation

The characteristic and QoS of MANET should be studied and tested to determine the QoS metrics required for voice traffic over MANET, such as delay and Energy. This can be done by reviewing the published work or by simulating the MANET using OpNet tools. Also, experimental investigating should be conducted on the AES encryption algorithm. The focus in this research will be on the encryption/decryption time and energy consumption in the nodes, to compare with newly proposed algorithms, in addition to security assessment for each algorithm, more details can be found in chapter 4.

3.5.1.2 Multi S-Box algorithm

Investigate and develop an encryption algorithm which aims to increase the complexity of the encryption process making it difficult to breach and at the same time don't increase the execution time and power consumption. The base of our work is the AES algorithm. A SubByte function using multi S-box transformation technique has been suggested to increase the confusion and complexity of encryption algorithm because each byte in state block will substitute with another Byte from S-Box table, as explained in chapter two.

So, the enhancements for AES functions are proposed. The AES algorithm has four functions: Sub-byte, Shift row, Mix column and Add round key. In this scenario, a new sub-byte transformation has been proposed. Sub-byte function enhancement aims to increase the security and the complexity of the AES algorithm and keeps the execution time approximately the same, by using multi S-box substitution method. The decryption has also been tested. All the details are explained in chapter 5.

3.5.1.3 Lightweight algorithm

Here a Lightweight and low energy encryption algorithm for voice over wireless networks are being developed and tested. The new encryption algorithm has to meet the QoS requirements of voice traffic and to be suitable for wireless devices. The Mixcolumn enhancement, in addition to reducing the number of rounds in AES algorithm processes, aims

to decrease the power consumption and execution time and to keep the security and complexity at the same level. The proposed algorithm employs similar methods with those used in the Advanced Encryption Standard algorithm (AES), with some changes and enhancements considering the limitations of wireless devices, this contribution can be found in chapter 6.

3.5.1.4 Triple Key Encryption Algorithm TKE

Finally, further development will be conducted for the algorithm proposed in chapter 5 and 6, to increase the security level by adding a 3rd key function. The SubByte function proposed in chapter 5 and mixcol proposed in chapter 6, in addition, the 3rd key function will be all used to propose the novel encryption algorithm for high security and lightweight consumption (see chapter 7). This approach has been already justified in previous sections and chapter 7 which contains more details.

The validation was carried out by implementing the theory in experiments. The validation approach adopts many ways and methods. A comparison with standard AES algorithm has been carried out and for more validation; we used two different processor specifications, to see how much power saving percentage for each processor.

3.5.2 Experiments

3.5.2.1 Experiment Tools

Microsoft Visual Studio 2015 has been used to build and write the proposed algorithm code using visual C++ programming language, which executes the encryption/decryption process. Console app was used to avoid any load on the processor and avoided additional delays. Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services, and mobile apps. The testing has been conducted in a laboratory environment using a High-performance lab computer with Microprocessor (quad-core i7, 8 GB RAM) running Windows 7, and the wireless laptop (Due Core, Ram 4 GB) was used for the other scenario. Using 2 different type of processors will help to measure the validity of the proposed scheme and prove the power saving percentage.

Furthermore, OPNET Simulator has been used to create and test Wnet. OPNET simulator is a tool to simulate the behaviour and performance of any type of network. The other technique used is parallel computing which helps to speed up the encryption/decryption process. OpenMP (Open Multi-Processing) is defined as an API that works on multiprocessing systems combined with shared memory platform. CrypTool 14.1 will be used to analyse the

encrypted data and to graph the statistical output. A Statistical Test Suite PRNG will be used as well for further security analysis of encrypted data. This toolbox was specifically designed for individuals interested in conducting statistical testing of cryptographic (P)RNGs. It was published by the National Institute of Standard And Technology (NIST).

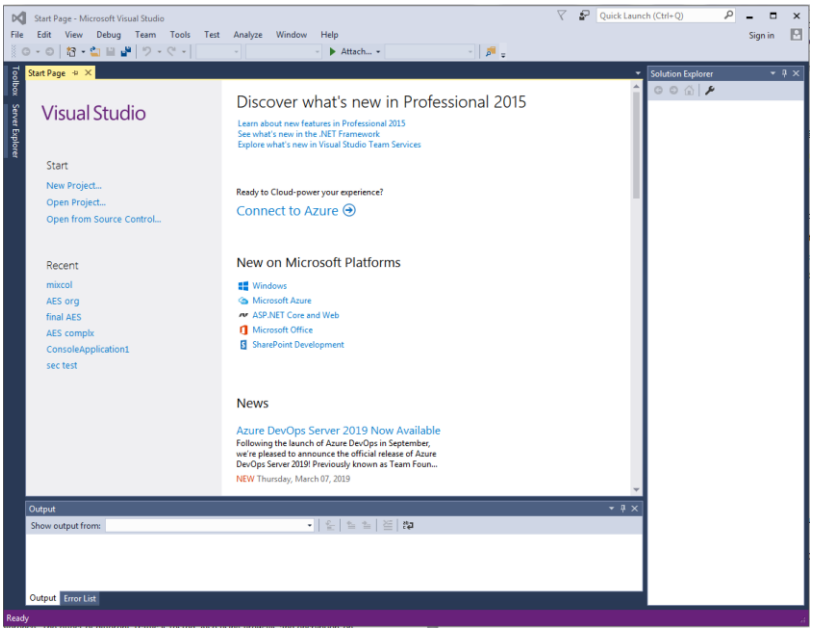


Fig. 3-3 Snapshot of Visual Studio

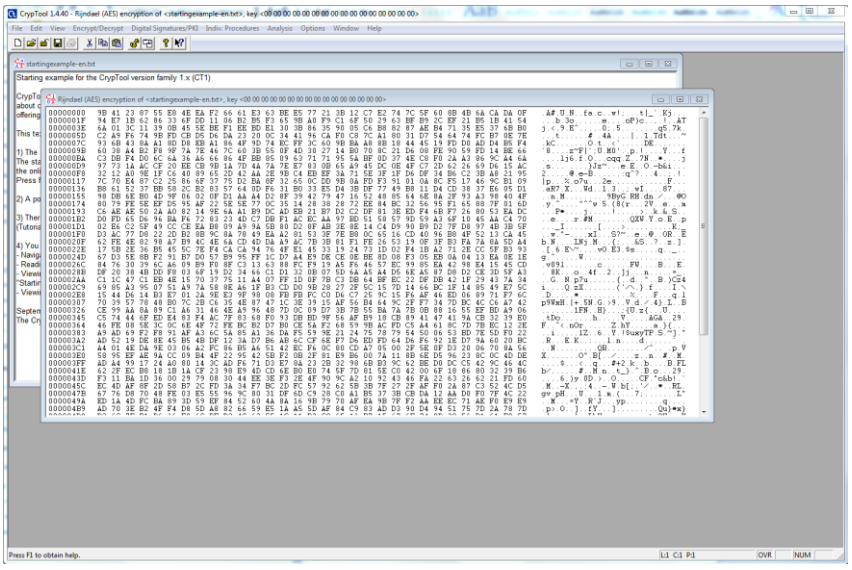


Fig. 3-4 Snapshot of CrypTool

3.5.2.2 File Attributes & Sizes:

The experiments and testing will be carried out on Audio file with .wav format and in different sizes, 128 KB, 540 KB, 1 M, and 1.48 MB and another size. The literature shows

that these sizes are common files used in network experiments (Ali, et al., 2014). The voice files used in this research are as following:

- Test file: it is a music voice with 128 KB size used in all the scenarios.
- Lecture file: it is a short talk by a lecturer with 540 KB size.
- Computer file: it is a conversation about the computer subject with 1.48 MB.
- Washing Machine file: a recorded sound for a washing machine while it working with different sizes.

Reading the audio files is not similar to reading data files, it is different. It needs to read the header first to know the characteristics of the file. So the header should be identified in the program and then start to scan the data to conduct the operation on it. Different file patterns have been used to check their effect like human voice and music and washing machine. We can also mention that the recording encrypted\decrypted files are easy to hear by human and it proves the successful implementation of experiment. The following code shows the audio file header reading process:

```
if ((err == 0) && (erw == 0))
   {
       fread(meta, 1, sizeof(header), Rfile); // read the source header file
       fwrite(meta, 1, sizeof(*meta), Wfile); //write header into destination file
              int flag = Check_Header(meta);
                  if (flag == 1)
                       {
                          printf("corrupted file");
                          exit(0);
                       }
```

3.5.2.3 Scenarios

The first step to investigate the traffic was implemented by Simulation scenarios of MANET over IP with VoIP and multimedia system to test the QoS. Opnet 18.5, used in this scenario. After that, the AES algorithm has been tested using visual studio 2015 in different devices to measure the time and energy consumed for encrypting the voice. Then, the proposed algorithms have been tested in the same way as AES testing.

3.5.3 Parameters Measured:

3.5.3.1 QoS Parameters

The most important parameters in this research are delay, Energy and file size.

- Delay: includes the delay time for packet from source to destination in the network. And the execution time spent to execute the encryption/decryption process. It is measured in second (sec).

- File size: of the audio file before and after encryption process which is measured in Byte (B), this size should be same for both plain and cipher so it is not going to affect the bandwidth of the network.
- Energy: consumed by the execution code to do the encryption/decryption process, measured in (Joule).

The algorithm's Energy consumption can be measured by calculating the total of computing cycles which are used in processes related to cryptographic tasks. For the calculation of the total energy cost of encryption, similar techniques are used as defined in (Jiehong & Detchenkov, 2016), based on the following equations:

$$\boldsymbol{B}\ cost\_encry\ (ampere\text{-}cycle) = \tau * I \qquad (1)$$

$$\boldsymbol{T}\ energy\_cost\ (ampere\text{-}seconds) = \frac{\boldsymbol{B}\ cost\_encry\ (ampere-cycle)}{F\ (cycle/sec)}$$

$$\boldsymbol{E}\ cost\ (Joule) = T\ energy\_cost\ (ampere\text{-}seconds) * V \qquad (2)$$

Where:
B cost_encry: a basic cost of encryption (ampere-cycle).
τ: the total number of clock cycles.
I: the average current drawn by each CPU clock cycle.
T energy_cost: the total energy cost (ampere-seconds).
F: clock frequency (cycles/sec).
E cost (Joule): the energy cost (consumed).

The energy consumption of cryptographic functions can be calculated, by using the cycles, the operating voltage of the CPU, and the average current drawn for each cycle (Jiehong & Detchenkov, 2016). For example, on average, each cycle consumes roughly 270 mA on an Intel 486DX2 processor or 180 mA on Intel Strong ARM.

In this research, the power factor produced from the above equation = 0.073

3.5.3.2 Security Parameters

The most important parameters for security analysis are:

- Entropy
- Binary Histogram
- Floating Frequency
- brute force attack
- Autocorrelation
- Randomness *P-value* and Poker test (All of these metrics will be explained in the following section)

3.5.4 Security Analysis

Appropriate security parameters are required to investigate the degree of randomness and encryption quality for the output file (binary sequences) produced by the proposed algorithm, statistical testing and mathematical measurements (Riad, et al., 2013). And then these metrics could be used to collect evidence whose output sequences are truly random and have a high encryption quality, which can be used safely in the converged network applications (Rana & Wankhade, 2017), (Jonge & Loo, 2013). The security analysis used in this research to confirm the security strength of the proposed schemes. There are three pieces of evidence to confirm the security of the encrypted files. The CrypTool graphs and numerical NIST tests results, in addition to the human recognition of the recorded sound, could be used as proof of successful encryption. The recorded sound (cipher) after encryption was completely unclear and nobody can understand it. There is a lack of security analysis in many research works. According to (Ye & Huang, 2016; Riad, et al., 2013), the security metrics are essential to measuring the security strength of any security system. Further details of the security metrics could be found in the following sections.

3.5.4.1 Cryptool 14.1

It used to analyse the encrypted data and to graph the statistical output. It is used to test the following parameters:

- **Entropy:** The entropy of a document represents an index of its information content. The entropy is measured in bits/ character and is defined by:

$$M[i] = \log_2 (1/p[i]) = -\log_2 (p[i])$$

Where M[i] (the information content of a message), p[i] is the probability, that the message M[i] is transferred by the message source. This means that the information content depends exclusively on the probability distribution with which the source generates the message. "*The information content of two messages chosen independently of one another equals the sum of the information content of the individual messages, with the aid of the information content of the individual messages, the average amount of information which a source with a specified distribution delivers can be calculated as the individual messages will be weighted according to the probabilities of their occurrence*" (Riad, et al., 2013).

$$\text{Entropy}(p[1], p[2], \ldots, p[r]) := -[p[1] * \log(p[1]) + p[2] \log(p[2]) + \cdots + p[r] \log(p[r])] .$$

The entropy of a source thus specifies its characteristic distribution for text which contains the character set (0 to 255), the entropy range between 0bit/char (for only one character in the text) and log(256) bit/char (for all 256 characters).

- **Binary histogram** expresses the frequency distribution of the characters of the document in graphical form. The distribution of each character should be different in both the plain and cipher. Thus, by using the cryptosystem, the cipher will not supply any useful information related to the plain-file. Then the differentiate cryptanalysis be more difficult.

Table 3-1 ASCII Code

Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex	
000	00		001	01		002	02		003	03	
004	04		005	05		006	06		007	07	
008	08		009	09		010	0A		011	0B	
012	0C		013	0D		014	0E		015	0F	
016	10		017	11		018	12		019	13	
020	14		021	15		022	16		023	17	
024	18		025	19		026	1A		027	1B	
028	1C		029	1D		030	1E		031	1F	
032	20		033	21	!	034	22	"	035	23	#
036	24	$	037	25	%	038	26	&	039	27	'
040	28	(	041	29	)	042	2A	*	043	2B	+
044	2C	,	045	2D	-	046	2E	.	047	2F	/
048	30	0	049	31	1	050	32	2	051	33	3
052	34	4	053	35	5	054	36	6	055	37	7
056	38	8	057	39	9	058	3A	:	059	3B	;
060	3C	<	061	3D	=	062	3E	>	063	3F	?
064	40	@	065	41	A	066	42	B	067	43	C
068	44	D	069	45	E	070	46	F	071	47	G
072	48	H	073	49	I	074	4A	J	075	4B	K
076	4C	L	077	4D	M	078	4E	N	079	4F	O
080	50	P	081	51	Q	082	52	R	083	53	S
084	54	T	085	55	U	086	56	V	087	57	W
088	58	X	089	59	Y	090	5A	Z	091	5B	[
092	5C	\	093	5D	]	094	5E	^	095	5F	_
096	60	`	097	61	a	098	62	b	099	63	c
100	64	d	101	65	e	102	66	f	103	67	g
104	68	h	105	69	i	106	6A	j	107	6B	k
108	6C	l	109	6D	m	110	6E	n	111	6F	o
112	70	p	113	71	q	114	72	r	115	73	s
116	74	t	117	75	u	118	76	v	119	77	w
120	78	x	121	79	y	122	7A	z	123	7B	{

124	7C	|	125	7D	}	126	7E	~	127	7F	
128	80	_	129	81	_	130	82	,	131	83	
132	84	"	133	85	...	134	86	†	135	87	‡
136	88	^	137	89	‰	138	8A	Š	139	8B	
140	8C	Œ	141	8D	_	142	8E	_	143	8F	_
144	90	_	145	91	'	146	92	'	147	93	"
148	94	"	149	95	·	150	96	–	151	97	—
152	98	~	153	99		154	9A	š	155	9B	›
156	9C	œ	157	9D	_	158	9E	_	159	9F	Ÿ
160	A0		161	A1	¡	162	A2	¢	163	A3	£
164	A4	¤	165	A5	¥	166	A6	|	167	A7	§
168	A8	¨	169	A9	©	170	AA	ª	171	AB	«
172	AC	¬	173	AD		174	AE		175	AF	¯
176	B0	°	177	B1	±	178	B2	²	179	B3	³
180	B4	´	181	B5	µ	182	B6	¶	183	B7	·
184	B8	¸	185	B9	¹	186	BA	º	187	BB	»
188	BC	¼	189	BD	½	190	BE	¾	191	BF	¿
192	C0	À	193	C1	Á	194	C2	Â	195	C3	Ã
196	C4	Ä	197	C5	Å	198	C6	Æ	199	C7	Ç
200	C8	È	201	C9	É	202	CA	Ê	203	CB	Ë
204	CC	Ì	205	CD	Í	206	CE	Î	207	CF	Ï
208	D0	Ð	209	D1	Ñ	210	D2	Ò	211	D3	Ó
212	D4	Ô	213	D5	Õ	214	D6	Ö	215	D7	×
216	D8	Ø	217	D9	Ù	218	DA	Ú	219	DB	Û
220	DC	Ü	221	DD	Ý	222	DE	Þ	223	DF	ß
224	E0	à	225	E1	á	226	E2	â	227	E3	ã
228	E4	ä	229	E5	å	230	E6	æ	231	E7	ç
232	E8	è	233	E9	é	234	EA	ê	235	EB	ë
236	EC	ì	237	ED	í	238	EE	î	239	EF	ï
240	F0	ð	241	F1	ñ	242	F2	ò	243	F3	ó
244	F4	ô	245	F5	õ	246	F6	ö	247	F7	÷
248	F8	ø	249	F9	ù	250	FA	ú	251	FB	û
252	FC	ü	253	FD	ý	254	FE	þ	255	FF	ÿ

To understand the Binary histogram figure, the above table can translate each number with the equivalent character. This ASCII table contains all 256 ASCII characters. The table contains four clusters of three columns apiece. The first column contains the decimal value (Dec.), the second column the hexadecimal value (Hex) and the third column the character itself if it is possible to display it on the screen.

- **Floating frequency** can be defined as the characteristic of local information content at separate points in the document. The floating frequency specifies how many different characters are to be found in any given 64-character long segment of the document (Riad, et al., 2013). And this should be diverse from Cipher.

- **Brute-force Test** it tries all possible keys to decipher the file. This test calculates the time required to recover the encryption key and decrypt the cipher.
- **The autocorrelation**: The autocorrelation of a sequence is an index of the match of different pieces of a sequence. The purpose of this empirical test of independence is to check the correlation between the binary sequence S and a version of S that has been displaced by t positions (ESSLINGER, 2008).

  Let t be a number, 1 <= t <= (n / 2) and fixed. The number of bit positions in *s* which do not agree with the version of *s* that has been displaced by t positions is determined by

  $$D(t) = \text{sum}[i = 0; i = n - t - 1]\ s_i \text{ XOR } s_{(i + t)}$$

  The test statistics used are given by

  $$X_5 = 2 * [(D(t) - [(n - t) / 2]) / (n - t) \wedge (1 / 2)]$$

  Where by X_5 approximates an N(0, 1) distribution, provided that n - t >= 10. As both small and large values for D(t) are unexpected.

  The autocorrelation function $C(t) = (A(t) - D(t))/n$.

3.5.4.2 A Statistical Test Suite

This toolbox specifically designed for individuals interested in conducting statistical testing of cryptographic (P)RNGs. It was published by the National Institute of Standard And Technology (NIST). The need for random and pseudorandom numbers has risen in many cryptographic applications. For instance, common cryptosystems employ keys that should be generated in a random fashion. Also, many cryptographic protocols require random or pseudorandom inputs at various points (Rukhin, et al., April 2010). A random, Random Number Generators (RNGs) and Pseudorandom Number Generators (PRNGs) explanation can be found in the appendix.

- **Test Description**

These tests use for analysing the random and Pseudorandom Number Generator for cryptographic applications. They are important and can measure the P_value for each bit stream have been chosen by the test. Furthermore, it counts the number of zeros and ones in each stream and calculates the result. The P_value can be calculated by using the threshold (alpha 0.01) When the bit streams result greater than alpha, then the test is passing (Rukhin, et al., April 2010). To compute the P_value, the following example illustrates the steps of it:

The zeros and ones of the input sequence (*e*) are converted to values of –1 and +1 and are added together to produce S_n. For example, if

e = 1011010101, then *n*=10 and $S_n = 1 + (-1) + 1 + 1 + (-1) + 1 + (-1) + 1 + (-1) + 1 = 2$

Compute the test statistic $S_{obs} = \frac{|Sn|}{\sqrt{n}}$, then $S_{obs} = .632455$

Compute *P-value* = ***erfc*** $\frac{|Sobs|}{\sqrt{2}}$ **,** where ***erfc*** is the complementary error function.

If the *P_value* ≥ 0.01, then the sequence is random. Otherwise, conclude it non-random.

- **System requirements**

This software package was originally developed on a SUN workstation under the Solaris operating system. All of the source code was written in ANSI C. In this research Oracle VM VirtualBox has run on window 10 with computer core i7 ram 8 GB and Ubuntu 17.1 has been installed in this VM.

Oracle VM VirtualBox: is a cross-platform virtualization application. It can be installed on current Intel or AMD-based computers. VirtualBox can produce and run a guest operating system (virtual machine) in a window of the host operating system.

Ubuntu: is an open source operating system for PCs. It is a Linux distribution based on the Debian architecture. It is commonly run on personal PCs and could be run on network servers.

- **Input and output file**

Data input may be provided in two techniques. If the user has a stand-alone program or hardware device which implements an RNG (Rukhin, et al., April 2010), the user may like to build as many files of random length as required. The files should have binary sequences stored as either ASCII characters containing zeroes and ones or as binary data where each byte contains eight bits worth of 0's and 1's. The NIST Statistical Test Suite can then independently examine these files. The output logs of empirical results will be stored in two files, *stats*, and *results*

- **Running the Test Code**

"The NIST statistical test suite can be run by type assess command, followed by the desired bit stream length, for instance, assess 1000. Then a sequence of menu prompts will be displayed. Then can select the data to be examined and the statistical tests to be applied". The first screen appears as follows:

GENERATOR OPTIONS

[00] Input File	[01] Linear Congruential
[02] Quadratic Congruential I	[03] Quadratic Congruential II
[04] Cubic Congruential	[05] XOR
[06] Modular Exponentiation	[07] Blum-Blum-Shub
[08] Micali-Schnorr	[09] G Using SHA-1

OPTION ----> 0
User Prescribed Input File: **data/data.pi**

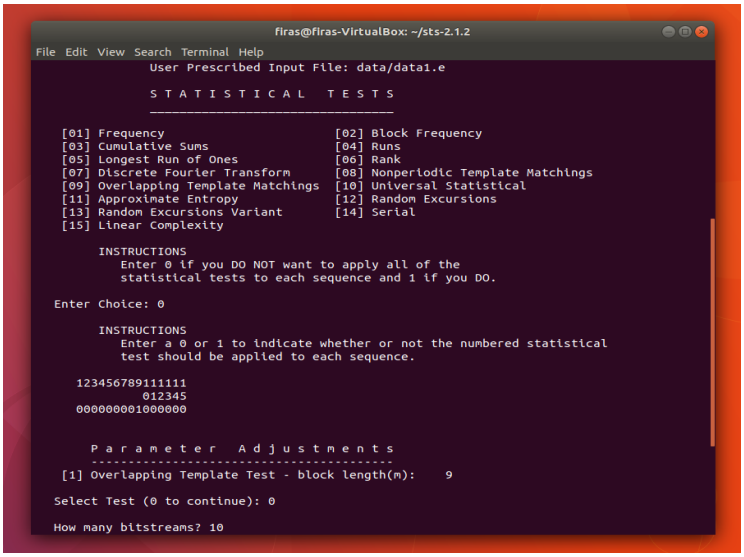


Fig. 3-5 Snapshot of NIST Test

3.6 Summary

This chapter describes the approach that was taken for the proposed design and its elements. Justification of research method and rationale for the research approach are explained. In addition to, the tools and the data have been used in the methodology. The details of how the results tested and analysed are also included. Furthermore, the whole research framework is graphically illustrated in this chapter.

4 Chapter four: Investigation of the Network & Encryption

4.1 Introduction

This chapter investigates and studies the network traffic and their requirements. Also, it experimentally investigates the AES encryption algorithm. It is assumed that cryptography processes occur in the nodes, so the focus will be on offline data. In the network environment, the end to end delay can be calculated as in fig below

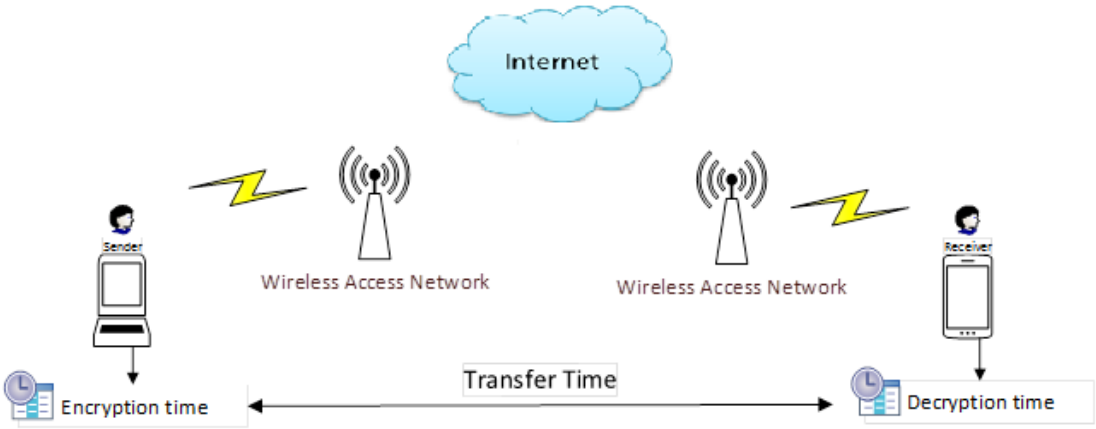


Fig. 4-1 Encryption and Decryption Time over Wireless Network

The focus of this research will be on the encryption/decryption time and energy consumption in the nodes. Because the transfer time is the same for both encrypted data or unencrypted data, as their size is still the same after both process (Wang, et al., 2014) and this will be proved in the experiment in this research. So the total time will be as follow:

$$T = Encryption\ time + Transfer\ time + Decryption\ time$$

Also, the energy consumed usually at the device battery (node). For these reasons, firstly, the traffic analysis in the wireless network should be conducted to determine which type of traffic should take extra care and to investigate the network behavior and requirement also the effect of nodes density could be measured in next section (4.2). Secondly, the standard cryptography algorithm then should be tested (offline) to measure its quality and to compare with the new proposed algorithm, section (4.3). Here in this research, the AES algorithm

has been chosen, as explained in the literature review. All of these tests are conducted in this chapter. This chapter is part of the published researches (Hazzaa & Yousef, 2017) and (Hazzaa, et al., 2019)

4.2 Characteristics & Experimental evaluation of real-time traffic in a Wireless network

4.2.1 Overview

This work simulates many types of traffic like FTP, Voice, and video conference. The results will help to decide which security methods could be used without affecting the quality of service. These simulation scenarios are conducted and testing data is analysed to test delay in each scenario, the results show significant differences

As explained in the introduction, the encryption process in networks environment takes place in the nodes, such as sender or router. It doesn't happen in transmission links between the nodes. So the delay time for transmitting the data between the sender and receiver is the same for both encrypted and unencrypted data because the size of them is still the same, as will be approved in the results. The encryption\decryption time is usually added to a total end to end delay between the nodes in the networks. In our work, the focus will be on the execution time at the node (wireless device) where the encryption process has been executed. Therefore, the focus should be on the execution time and power for encryption process needed in the node (device).

Study of the characteristics of real-time traffic and their requirements in wireless networks is very important before any security method is proposed. For instance, performance measures such as delay, throughput, and network load should be carefully checked for different traffic types such as voice, video, and text. This will enable decision making on the selection of security methods which could be used without affecting the QoS for these networks. According to (Albonda, et al., March 2017) a distinctive difference in voice packet delay can be noticed, when more than 35 nodes participate in the MANET. Figure 4.2, shows how each mobility model affects the delay in different ways and in relation to the number of nodes in the MANET cluster. It is clear that more delays have been added when increasing the participated nodes, in addition to the mobility speed.

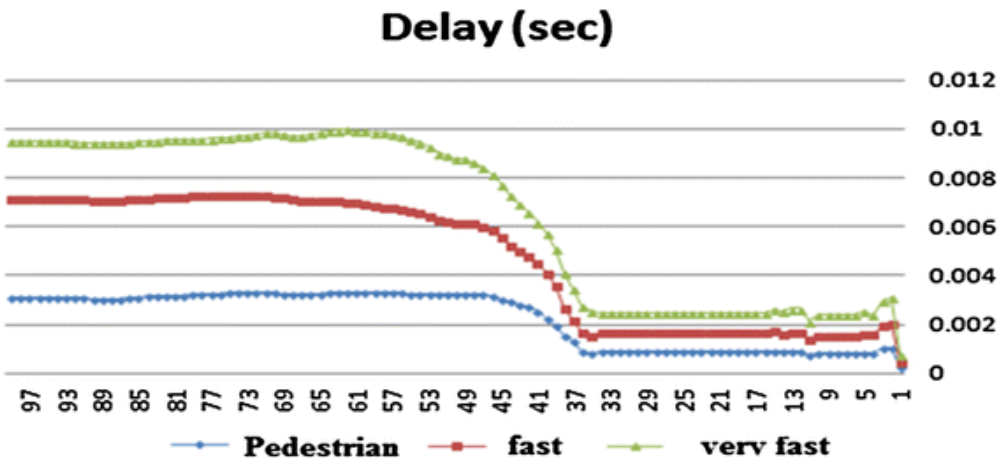


Fig. 4-2 Packet delay (in sec) versus the nodes volume (Albonda, et al., March 2017)

In the proposed scheme, it is essential to consider these sensitive requirements for real-time traffic and it should make the tradeoff between the security level and QoS requirements. And it should offer to the network designer a range of cryptography options, depending on their network density and traffic requirement. For instance, in the small network, it may use the encryption algorithm with 20% energy saver, while in the large wireless network it urgently needs more than 50% energy saver (i.e WSN). So the any proposed algorithms should consider these facts. In this work, three encryption algorithm will be introduced with different range of energy saving to meet the new requirements of the wireless networks.

4.2.1.1 Security and QoS Relation

As mentioned, Real-time applications over wireless networks often suffer from jitter, delay, and packet loss. Due to the recent increase in wireless mobile users and data usage, the available spectrum is limited and there is a need to devise new methods (Hazzaa, et al., 2019). Also, implementing security enforcement in wireless networks add more load to the traffic and hugely affect the quality of service metrics. As explained before, the cryptography is the most important principle in cybersecurity, and it makes the data unreadable, however, it also consumes a lot of QoS metrics. Therefore, it is important to find some way to address these issues. The first thing is how to choose the suitable method and this can be done by a traffic study. Traffic classification is very important for any researcher in a network area because it would help them to understand the characteristics of each type of traffic; this will help to know the requirement of each type in term of QoS (Liang & Chao, 2011) (Hazzaa & Yousef, 2017). As we know, there are many differences between real-time traffic and other traffic. In real time traffic, there is a huge requirement for Quality of Service metrics such as

Voice and Video Conference, while it is less important in another kind such as FTP. As mentioned, understanding the characteristics of each type of traffic will help to know the requirement of each type in term of quality of service QoS. For example, when need to encrypt the data; the quality of service (QoS) metrics should be considered because each type of traffic has some specific requirements, like the delay in real-time traffic. In this work, many types of traffic like FTP, Voice and video conference has been simulated and many results has been obtained which will help to decide which security methods could be used without affecting QoS.

4.2.1.2 QoS Metrics Specification for MANET

One of the main challenges in integrating WSNs to the Internet is to ensure reliable traffic flows between both networks, to offer an end-to-end quality of service (QoS) (Acharyya, et al., 2017). For instance, "*Communication between vehicles enables a wide array of applications and services ranging from road safety to traffic management and infotainment. Each application places a distinct quality of service (QoS) constraint on the exchange of information. The required performance of the supported services differs considerably in terms of bandwidth, latency, and communication reliability. For example, high-bandwidth applications, such as video streaming, require highly reliable communication*" (Brahim & Mir, 2017). The aims of this section are to design the MANET and to determine the QoS requirements for MANET and to study the behaviour of different type of traffic in this network.

4.2.2 Investigating the Performance and QoS metrics

The aim of this test is to find out the delay and throughput of a different kind of traffic in MANET with and without connection to the Internet. This work is part of the published research in (Hazzaa & Yousef, 2017). The parameters were chosen according to (Albonda, et al., March 2017) All the tests and the results are available in the appendix B.

4.2.3 The Effect of Nodes Density on Real Time Traffic

In this work, the Ad hoc network with a different number of nodes with voice traffic has been simulated, and obtained many results that will help to see the effect of nodes density on QoS. The same parameters as in the previous sections are used. And two scenarios are carried out, scenario1 with 50 nodes and scenario2 with 25 nodes as in (Albonda, et al., March 2017). The traffic type that is passed through both networks is Voice. The similar parameters

have been configured as in the section (4.2.2). This work is part of the published research in (Hazzaa, et al., 2019)

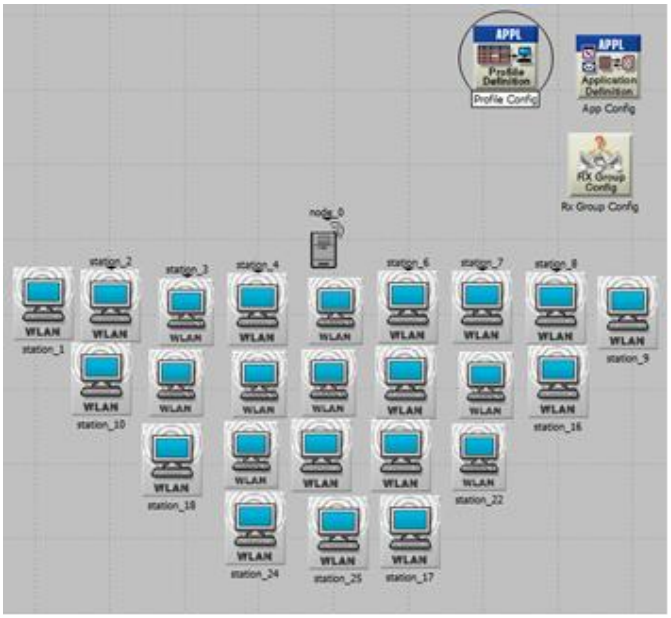


Fig. 4-3 Ad hoc Network with 25 nodes to Test the Voice Traffic

The result showed a clear difference between the scenarios and the effect of nodes density on the voice traffic in the network.

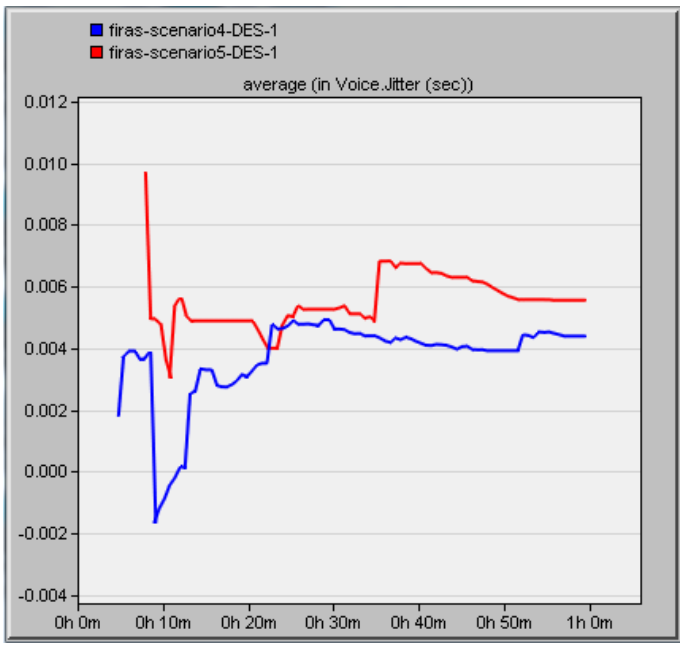


Fig. 4-4 Voice Jitter in the both Networks

Figure (4-4) illustrates the result of the jitter (sec) in the designed networks for two scenarios. As we can see there is a difference in the result between the two scenarios regarding the

number of nodes. For 50 nodes network, the average is approximately (0.006s) while for 25 nodes it reaches near (0.004s). The red line represents the delay of 50 nodes

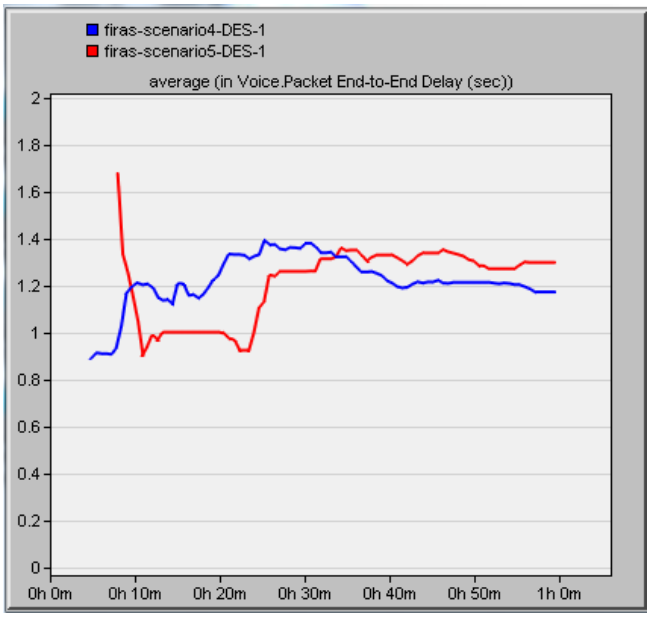


Fig. 4-5 End to End Delay

Figure (4-5) shows the result of End-to-End delay (sec) in the designed networks for two scenarios. As we can see there is also a difference between the two scenarios regarding the number of nodes. For 50 nodes network, the average is approximately (1.4s) while for 25 nodes it reaches near (1.2s).

It is clear from the above figures and previous sections; there are many differences among the QoS parameters during the increasing of participated nodes. For real-time application and wireless environment, this is very crucial. For example, the delay increases with the increasing of node density, in addition, this delay wills double when implementing security or encryption processing. So it is vital to planning the network requirements and its limitations before implementing any security policy.

4.3 Investigating the Standard AES algorithm

As explained in chapter two, AES is Symmetric-key cryptography and refers to encryption methods in which both the sender and receiver share the same key. It is fast and flexible; it can be implemented on various platforms especially in small devices (Ayyappadas, et al., 2014). Also, AES has been carefully tested for many security applications. In October 2000, the National Institute of Standards and Technology (NIST) publicized the Rijndael algorithm, which is selected for the Advanced Encryption Standard

(AES). The symmetric key AES algorithm is considered the most common and widely used at present to send information in a secure manner (NIST, 2004). It is the strongest algorithm in information security and it highly resists different types of attacks. This section, experimentally tests the standard AES algorithm to investigate their parameters such as time, power and security strength and to compare it with the results of the proposed algorithm.

4.3.1 Testing

This test aims to apply the encryption/decryption process on the audio file using AES algorithm, to determine the execution time and Energy consumption. Firstly, the program should read the audio file and then execute the cryptography process. Reading the audio files is not similar to reading data files, it is more complicated; it needs to read the header first to know the characteristics of the file. Therefore, the compiler should identify the header in the program and then start to scan the data to do the operation on it.

The header of the file contains many variables as:

```
struct header_file
{
	char chunk_id[4];
	unsigned long chunk_size;
	char format[4];
	char subchunk1_id[4];
	unsigned long int subchunk1_size;
	short int audio_format;
	short int num_channels;
	unsigned long int sample_rate; //sample_rate denotes the sampling rate.
	unsigned long int byte_rate;
	short int block_align;
	short int bits_per_sample;
	char subchunk2_id[4];
	unsigned long int subchunk2_size; subchunk2_size denotes the number of samples.
} header;
typedef struct header_file *header_p;
```

These identify the file type and file format, which is .wav format in the test. These variables represent the first 44 Byte in audio file which describe the voice characteristics.

The testing was carried out using Visual Studio 2015 with C++ programing language in Lab's laptop computer with processor Quad Core i7, Ram 8 GB. And the original AES algorithm has been tested on an audio file with different sizes. The audio file stored in the computer in drive C and the path of the file as follows:

C:\Users\Fih102\Desktop\AES org\AES org

Fig. (4-6) shows the running of the Test for voice file test.wav after implementing the standard AES algorithm.

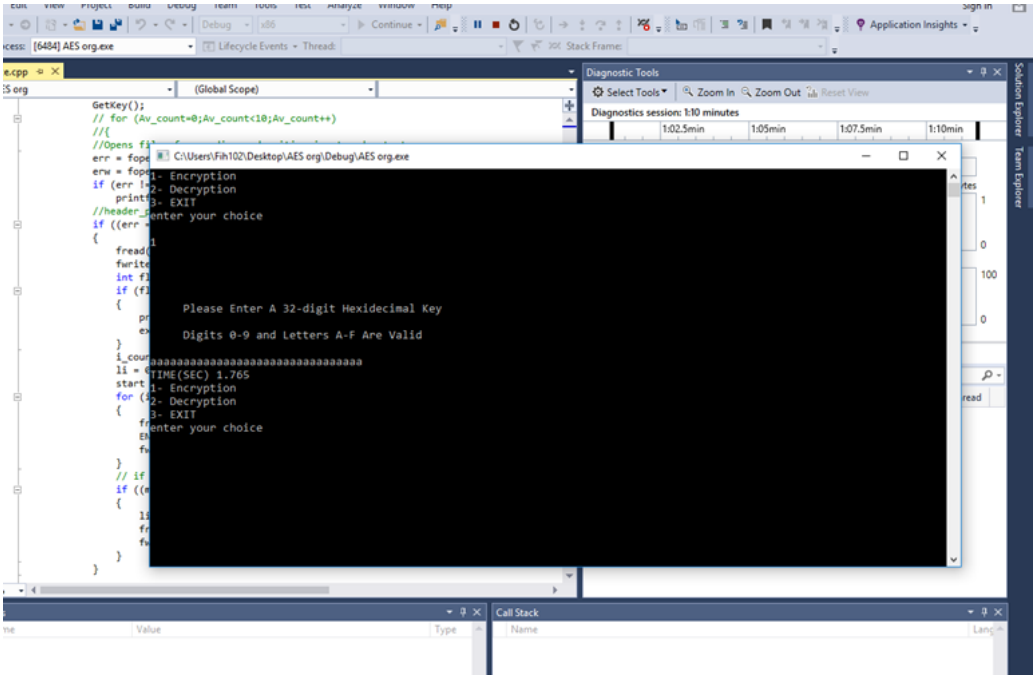


Fig. 4-6 Snapshot

As explained in the literature, standard AES using a single fixed S-box table in SubByte transformation function. The following table has been used in this test.

Table 4-1 S-box

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

After executing the encryption program (source.cp) by the software, the command line appears and asks to enter the main encryption key. The key used in this test is:

Main Key = 123456789abcdef123456789abcdef12

The length of the key is 32 char, means 16 bytes.

After entering the key and hitting enter, the execution start to encrypt the file. The choosing of the file depends on the following instruction:

```
//Opens files for reading and writing input and output.
          err = fopen_s(&Rfile, "test.wav", "rb");
          erw = fopen_s(&Wfile, "AESOUTP.wav", "wb");
```

For the decryption the instruction will be as:

```
          err = fopen_s(&Rfile, " AESOUTP.wav", "rb");
          erw = fopen_s(&Wfile, "finalOUTP.wav", "wb");
```

So, the output file in encryption process will be the input file in decryption process.

The following code executes the encryption process on the input audio file:

```
for (i = 0; i<i_count; i += 16)
        {
                fread(State, sizeof(char), 16, Rfile);
                ENCRYPT();
                fwrite(State, sizeof(char), 16, Wfile);
                        }
  if ((meta->subchunk2_size % 16) != 0)
       //if the length less than 32 no padding the values stores as it is
                        {
  li = meta->subchunk2_size - i; //to get the number of bytes that are less of 32
                fread(Temp, sizeof(char), li, Rfile);
                fwrite(Temp, sizeof(char), li, Wfile);
    }
```

The execution of this test has been conducted several times to ensure the accuracy of the results. The average of the final result has been calculated as illustrated in the result section.

4.3.2 Results

The results below show the execution time and energy consumption which consumed by the standard AES algorithm for both the encryption and decryption processes.

Table 4-2 Encryption Time

File Size	Pattern	Encryption Time (Sec)	Decryption Time (Sec)
128 K	Human voice	0.477	0.462
540 K		1.493	1.387
1 M		2.787	2.668
1.48 M		4.078	3.885

Table 4-3 Encryption Energy

File Size	Pattern	Encryption Energy (µJ)	Decryption Energy (µJ)
128 K	Human voice	0.035	0.034
540 K		0.109	0.101
1 M		0.2	0.195
1.48 M		0.298	0.284

Table 4-4 Different Pattern

(a) Time

File Size	Pattern	Encryption Time (Sec)	Decryption Time (Sec)
128 K	Music	0.472	0.459
540 K		1.485	1.383
1.48 M		4.018	3.835

(b) Energy

File Size	Pattern	Encryption Energy (µJ)	Decryption Energy (µJ)
128 K	Music	0.034	0.034
540 K		0.108	0.101
1.48 M		0.293	0.280

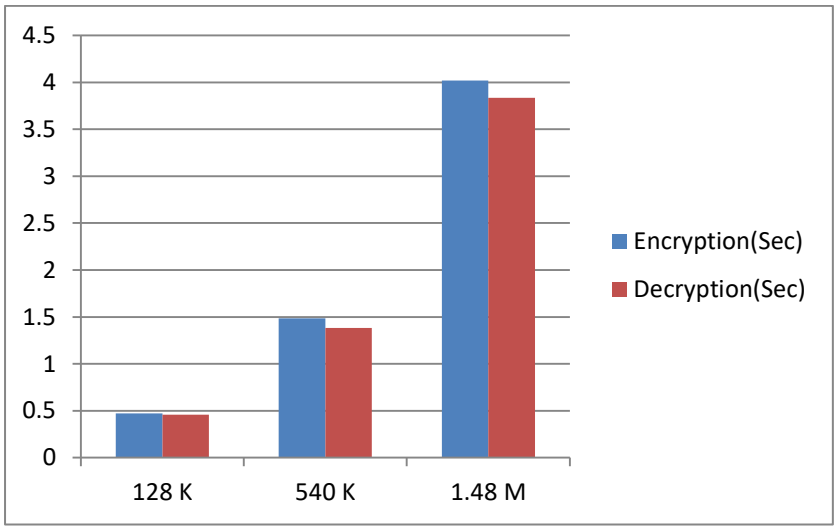
Table (4-2) illustrates the amount of the execution time which has been taken by the standard AES algorithm to encrypt and decrypt the audio files with different size. The highest level it reached nearly 4.078 sec for 1.4 MB while the lowest is 0.477 sec for 128 KB file size. The sample rate of each file has also been mentioned in these tables.

Table (4-3) shows the energy consumption which has been consumed by the standard AES algorithm, it has been calculated as described in chapter three. The highest level for both the

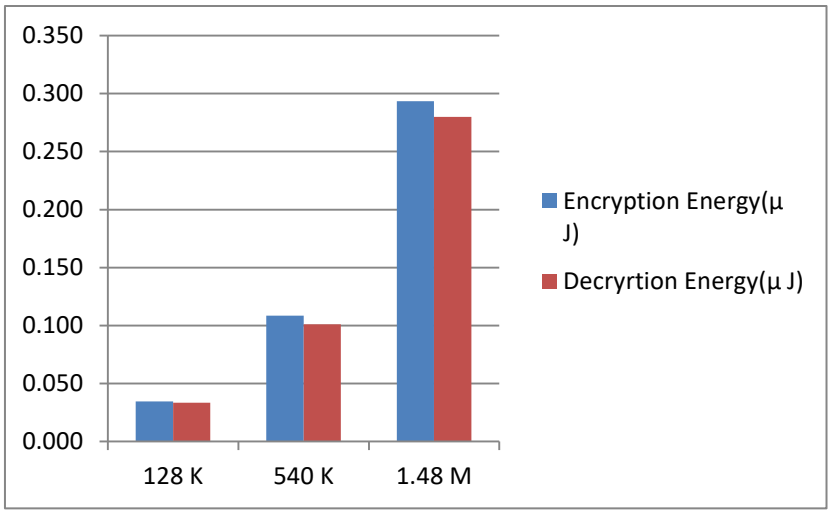
encryption and decryption processes are reached approximately 0.298 µ Joule and 0.284 µ Joule, respectively. While the lowest is 0.035 µ Joule for 128 KB file size.

Table (4-4) shows the results for the same parameters but with different voice pattern. It is clear that similar values have been obtained from both tests.

The statistics figure below show the characteristic of encryption and decryption process in the AES algorithm, it is clear that the same amount of time for both processes (encryption\decryption).



(a)



(b)

Fig. 4-7 Results

4.3.3 Security Analysis

The AES algorithm has a high-level of security, as explained in chapter two. However, this section conducted an investigative security analysis to ascertain its strength and to compare this analysis with the proposed algorithm analysis in the following chapters. So the security parameters have been tested in this analysis such as Binary histogram, frequency test, and entropy. All these parameters have been explained in chapter three.

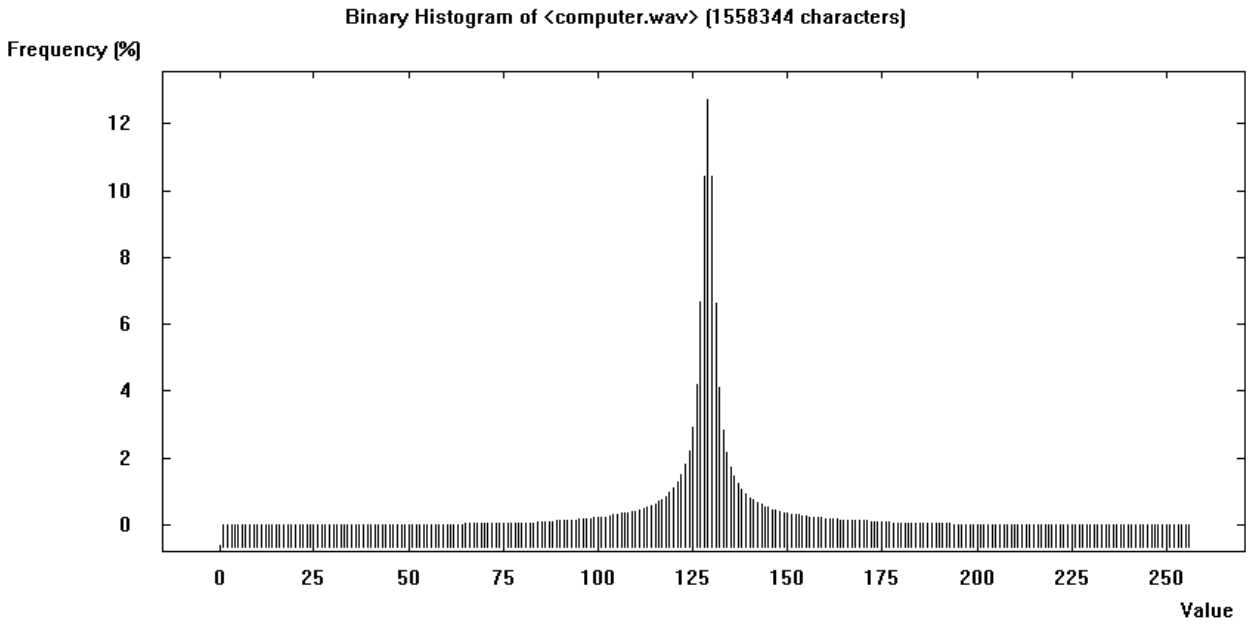
Table (4-5) shows the Entropy test result for the encrypted file in the AES standard. There is no doubt that the standard AES algorithm showed a good value, which was 7.99 from the maximum possible value of = 8.

Table 4-5 Entropy

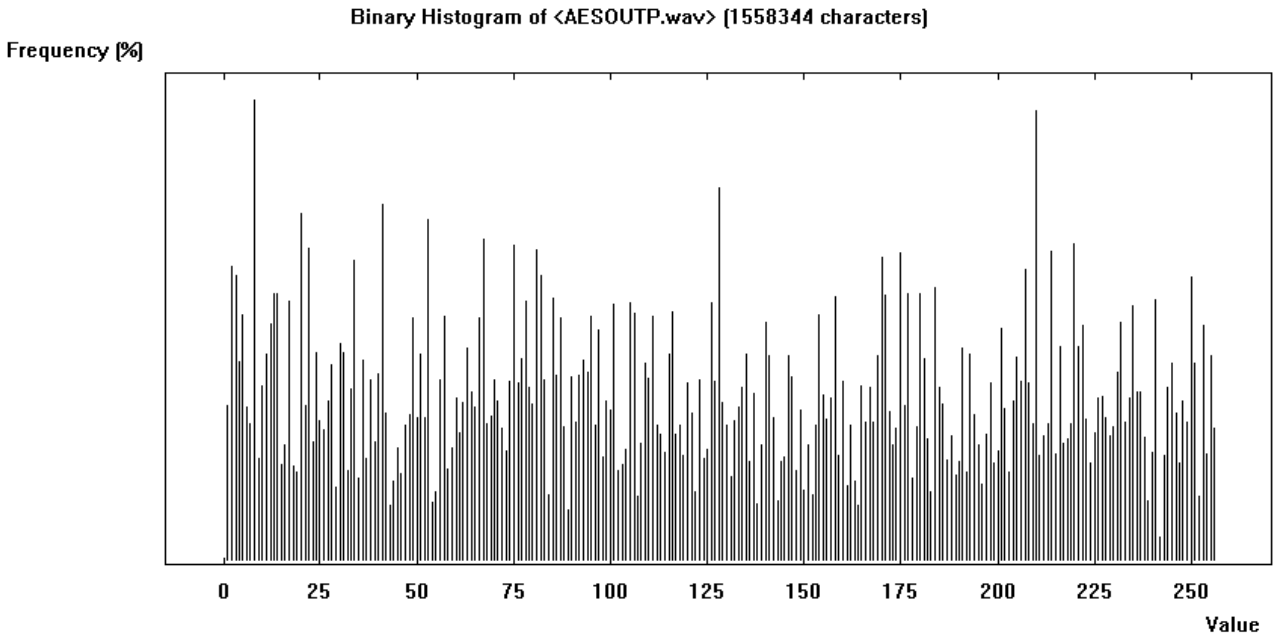
Audio file	**Plain**	**AES cipher**		
	Entropy	Entropy	Max. possible Entropy	Possible byte value
test	7.79	7.99	8	256
teaching	5.65	7.99	8	256
washing	5.4	7.99	8	256
computer	5.13	7.99	8	256

The figures below show the security analysis for the encrypted file for standard AES algorithm. Fig.(4-8) represents the Binary Histogram of the original audio file (computer.wav) before the encryption and the Cipher file. It is clear that a huge difference between two figs and a good binary distribution has achieved after the encryption.

Also, figure (4-9) illustrate the floating frequency of both the plain and cipher file encrypted by AES. In addition to figure (4-10) describe the autocorrelation for the cipher by AES algorithm after the test. These results will be compared with the proposed schemes to show the new enhancements.

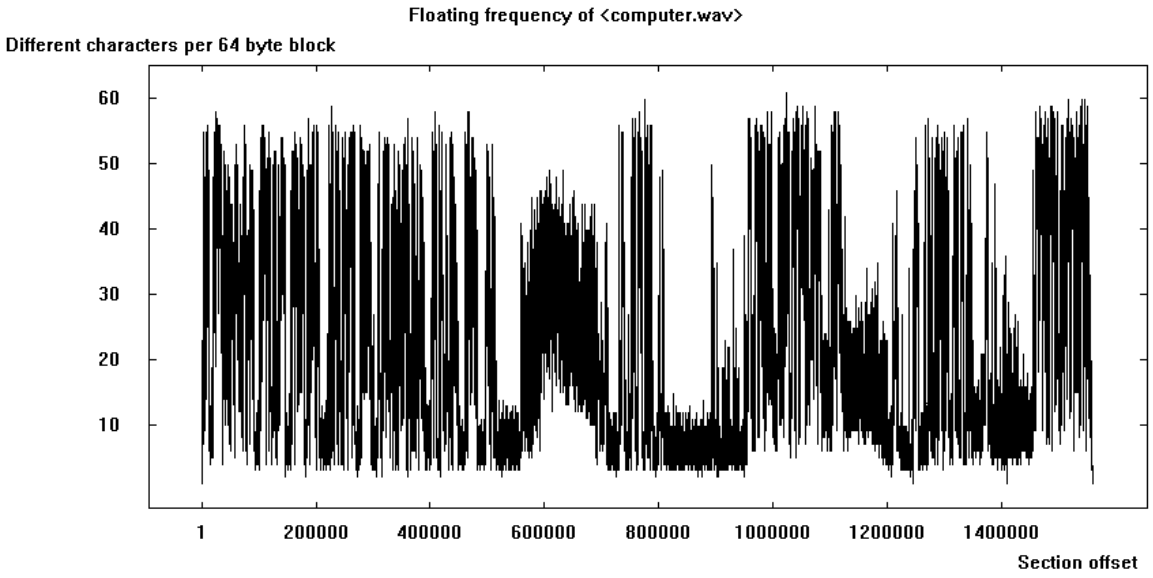


(a) Plain audio file

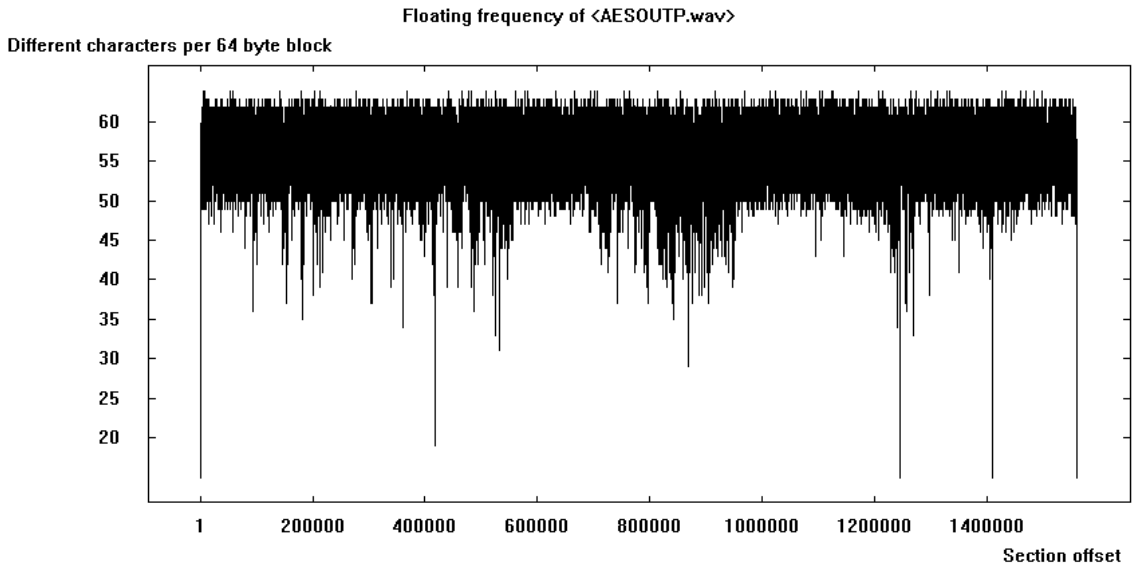


(b) Cipher audio file by AES

Fig. 4-8 Binary Histogram



(a) Plain



(b) Cipher

Fig. 4-9 Floating Frequency

It is clear from the figure (4-9) that the floating frequency for the encrypted file is random, and the most of frequency is between 50 – 60 different characters per 64-byte block compare with fig which ranged between 10 and 55. This means that there is significant randomness in the standard AES algorithm, keeping the diffusion in the cipher and leading to more complexity in the relationship between the cipher and the plaintext. Therefore, according to the literature review (Stallings, 2017) and this test, the standard AES is considered secure.

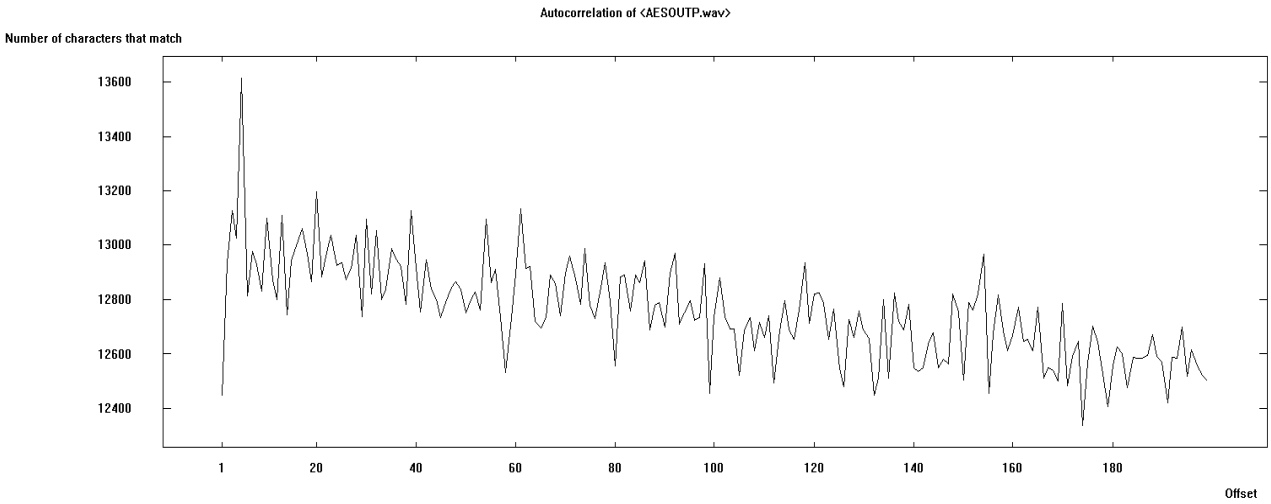


Fig. 4-10 Autocorrelation

In addition to the previous analysis, some tests have been carried out, such as poker test and brute-force attack, to test its strength. These tests have been carried out on the encrypted audio file and both of these tests were passed, as shown in Fig (4-11)

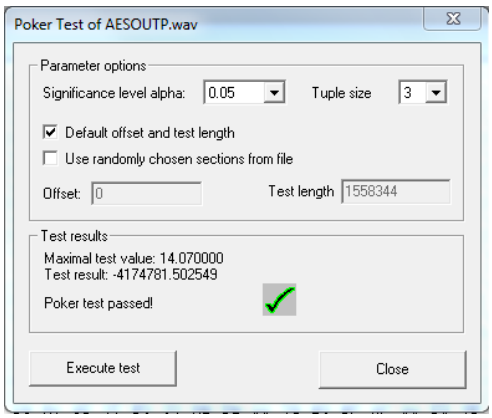


Fig. 4-11 Poker Test

4.4 Discussion & Conclusion

In addition to the literature review, this chapter provided a clear understanding of the network requirements and its behavior. The importance of the knowledge of characteristics of each type is useful because it will help to understand the QoS metrics for each type of traffic. Also, this chapter showed the execution performance of the standard AES algorithm and its strength.

All experiments were conducted in the offline scenario, because as mentioned before the encryption/decryption process occurs in the node itself, so don't worry about the transfer of

the data through the network after encryption. The only thing we should care about is the size of cipher which must be the same as in the plain text.

The investigation of the standard AES algorithm showed its performance, in term of execution time and energy consumption, in addition to the security analysis of the encrypted output. These investigations will help to compare it with the proposed schemes to validate and evaluate their results. So the objective of this chapter has been achieved in a good manner.

AES is very secure because it uses substitution, permutation, mixing, and keys, in addition to many rounds of iterations. These operations offer the confusion and diffusion needed to protect any cipher from cryptanalysis attempt. So, any proposed cryptosystem based on AES features will be efficient and secure. In this research, the proposed crypto-schemes based on AES features because of its strength.

The next chapters will propose the new cryptography algorithms which will meet the QoS requirements of the voice, and to reduce the encryption cost for current AES algorithm execution, as stated above in this chapter.

In the proposed scheme, it is essential to consider these sensitive requirements for real-time traffic and it should make the tradeoff between the security level and QoS requirements. And it should offer to the network designer a range of cryptography options, depending on their network density and traffic requirement. For instance, in the small network, it may use the encryption algorithm with 20% energy saver, while in the large wireless network it urgently needs more than 50% energy saver (i.e WSN). So the proposed algorithm should consider these facts.

5 Chapter five: Multi S-box Encryption Algorithm

5.1 Introduction

As explained in previous chapters, Wireless Mobile ad hoc network connectivity to the Internet (WMANET-IP) is facing huge security challenges because every node can enter and leave without central admission or authority. So a huge danger comes from anywhere in the network (internal and external). Encryption of data in WMANET is very important to protect the confidentiality of the data and it is the key principle of cybersecurity. It helps to cipher the information and makes it unreadable. However, implementing the current encryption methods is a big issue for this kind of networks, especially with real-time traffic, because of the Energy limitation of nodes in WMANET. Therefore, these methods should be complex, execute quickly, and power saver (do not consume a lot of battery Energy). All of these points will keep the security at a high level and keep the QoS at an acceptable level. There are two factors that can help to increase the complexity of any cryptography algorithm, which is the Confusion and Diffusion, the strong encryption needs much more confusion and diffusion.

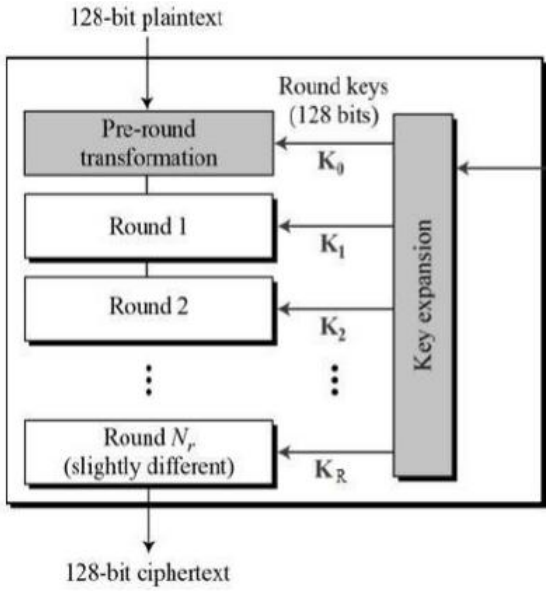


Fig. 5-1 Encryption Process (Stallings, 2017)

This chapter is going to investigate and develop an encryption algorithm which aims to increase the complexity of encryption process making it difficult to breach and in the same time don’t increase the execution time and power consumption. This chapter is the first stage

in the proposed framework. The output of this chapter will be used to propose the main algorithms in chapter 6 and 7, so more analysis and evaluation could be found in these following chapters. The base of this work is the AES algorithm. A SubByte function using multi S-box transformation technique has been suggested to increase the confusion and complexity of encryption algorithm. Also, further techniques will be proposed in this chapter. Many recent types of research have been introducing such technique which can solve the fixed structure issue in AES algorithm, (Abhiram, et al., 2015) claim that Static S-Boxes are implemented using look-up tables which will never vary with the input text or input key. This technique makes reverse engineering very simple for the purpose of cryptanalysis. Thus it is essential to generate S-Bytes at run time. It is beneficial if the S-byte generated during run time varies with the input key. (Alamsyah, et al., 2017) built S-box using a basic polynomial equation and the addition of a constant 8-bit vector different from the standard AES**.** (G & S, 2016) proposed a new technique to generate S-Box dynamically which will intensify the complexity of S-Box construction to encounter any possible attack on the fixed S –Box. (Ali, et al., June 2014), (Mohammed & Rohiem, 2009) introduced a multi S-box mapping technique which helped to increase the complexity of the algorithm. However, the author did not consider quality requirements. Also, there is a lack of convincing security argument which proves its strength. This chapter will address these gaps by testing and analysing the new encryption algorithm.

Aims

- To propose and investigate an encryption algorithm with a high level of complexity and at the same time keeping the execution time and power consumption at the same level.
- Address the fixed structure of SubByte transformation function, to increase the confusion in the cipher.
- Propose further flexible techniques using the dual key for the encryption process to increase the security.
- Analyse the security parameters to prove their strength.

Methodology

In this chapter, a quantitative research method has been adopted in which involves running security encryption experiments for audio files with different sizes. A proposed algorithm has been explained and been tested. The delay time and Energy consumed parameters have also been measured. A security analysis has been conducted to prove the algorithm strength. The testing was carried out using Visual Studio 2015 with C++ programing language. Four scenarios have been conducted in Lab's wireless laptop and computer with window 7 using a different kind of processer. In the first scenario (quad-core i7, Ram 8 GB) used. The other scenario (Due Core, Ram 4 GB) and the proposed Multi S-box algorithm have been tested on an audio file with different size. Finally, the evaluation and comparison with the standard algorithm have been carried out.

A security analysis has been conducted to test the security level of the new algorithm. Many security parameters have been measured, such as the randomness of the encrypted data like Entropy, Histogram, and Floating frequency test. The poker test and frequency test are carried out as well. The CrypTools 1.4 for cryptography and cryptanalysis have been used to carrying out these tests. All the tests have conducted on audio files.

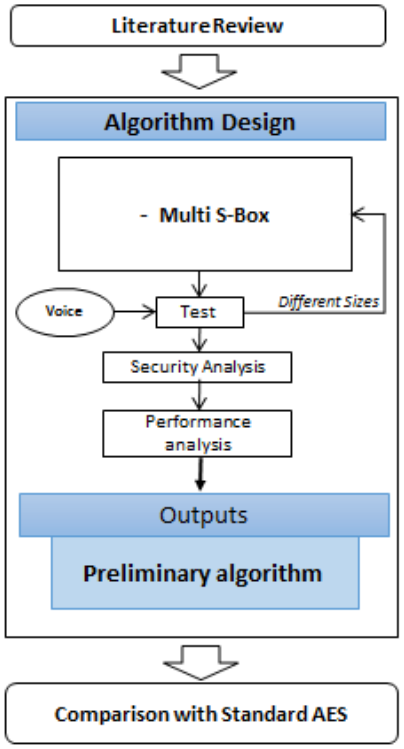


Fig. 5-2 Stage 1 Methodology Design

5.1.1 Finite Field Arithmetic

A finite field also called a Galois Field (GF) is a field with only finitely many elements. The finite field (GF 2^8) consists of the 2^8= 256 different numbers from (0 ... 255) represented by one byte eight bits. Special XOR – and modulo-operations make sure that the sum and the product of two finite-field elements remain within the range of the original finite field (8-bit) (Bucholz, 2001).

5.1.1.1 Byte Representation Forms

There are four different representation forms could represent the byte numbers, in this section, we will explain each kind with the examples:

Binary Representation: in this type, the byte could be represented by 8 – bit, as in the following example:

$$10100011_b$$

Decimal Representation: the above binary number can be represented by multiplying every bit by its corresponding power of two:

$$10100011_b = 163_d$$

Hexadecimal Representation: The numbers 0 ... 15 can be expressed by a group of four bits called a nibble. The numbers 10 ... 15 cannot be represented by a single decimal digit (0 ... 9) and are, therefore "abbreviated" by the letters A ... F in hexadecimal notation (Bucholz, 2001):

$$1111_b = 15_d = F_h$$

$$10100011_b = 163_d = A3_h$$

The polynomial representation: this is a method can be performed by multiply each bit by X. the advantage of using this method is to represent the numbers in GF. Fig. (5-3) illustrates the multiplication of two binary numbers.

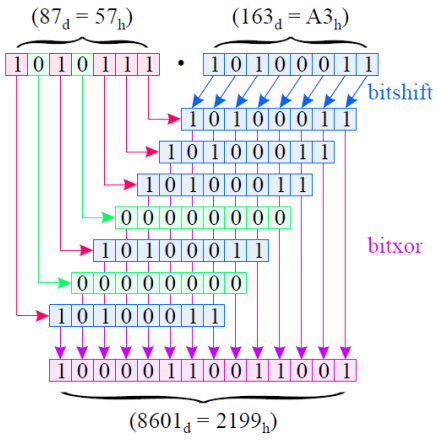


Fig. 5-3 Binary Multiplication (Bucholz, 2001)

Unfortunately, the resulting polynomial has a degree greater than 8 bit, can therefore not be expressed in one byte, not a GF(2^8), so it has to be transformed back into the ”byte range” by the modulo division.

5.1.2 Substitution Transformation in AES

SubBytes transformation is a substitution function which is considered as a nonlinear transformation. Each byte in the State matrix is working independently with remapping their values of the S-Box table. The S-Box is created using the multiplicative inverse in the finite field GF(2^8), S-boxes are crucial components of many cryptographic primitives (Alsalami, et al., 2016). If such components are removed, these primitives can be easily broken by performing linear analysis to the inputs, outputs and the secret key assuming that other components are nonlinear. The secret key bits can be easily realized from the input and output bits using certain linear algebra methods like the Gaussian elimination. Therefore, it is vital that the S-boxes used in any cryptographic basic are nonlinear and very well created against linear attacks (Alsalami, et al., 2016).

5.1.2.1 S-Box generation

Encryption/decrypting functions cipher and inv_cipher using the substitution tables (s_box) and (inv_s_box) to directly substitute a byte GF(2^8) by another byte of the same finite field.

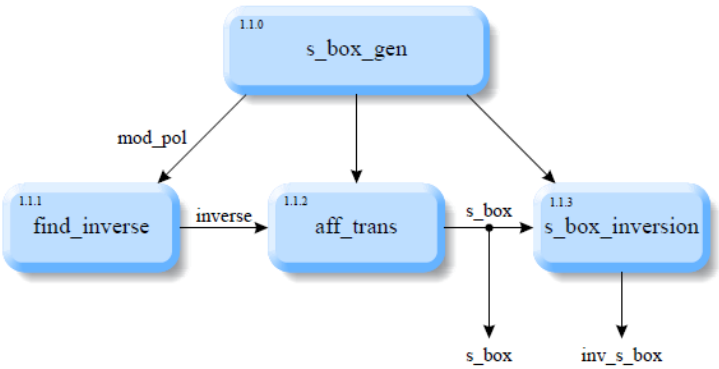


Fig. 5-4 S-Box generation (Bucholz, 2001)

find_inverse is the first step in the S-box generating process is to search for the multiplicative inverses of all elements of the finite field GF(2^8). Or, for all possible 256-byte values b, find the byte b−1 that satisfies (Bucholz, 2001)

$$b*b^{-1} = 1$$

*Where * represent a polynomial multiplication defined in poly_mult.*

After that, the second stage is an affine transformation, which involving of a polynomial multiplication with a specific constant (31_d = 00011111_b) modulo another constant (257_d = 100000001_b) and the XOR addition of a third constant (99_d = 01100011_b):

$$b_{out} = b_{in} \bullet 31_d \bmod 257_d + 99_d$$

where + bit-wise XoR operation.

5.2 Proposed algorithm

5.2.1 Implement SubByte Transformation using Multi-S-boxes

As explained in chapter two/section (2.9.3), the Advanced Encryption Standard AES algorithm consists of four functions; here the first function SubByte will be modified to increase their strength. In this section Multi, S-box encryption has been proposed. The goal of the proposed method is to use multi S-box in the transformation process for SubByte function using dual keys, instead of fixed structure for the S-Box used in the AES Rijndael. Similar techniques as in (Ali, et al., June 2014) are used. The goal is to make the new algorithm more secure without affecting the running cost, so it is very important to increase the complexity level in SubByte layout since the other AES functions will modify to achieve the tradeoff between the security and the time and power consumption. The proposed SubByte function solves the problem of the fixed structure which will lead to the generation of more secure block ciphers. Each byte in the State matrix will be encrypted using different S-Box tables created by the first key, and this, in turn, increases the security of the AES block cipher system. The key benefit of the proposed function is that a huge number of S-Boxes could be generated.

The second key represents a random distribution of the S-Boxes created by the first key. This key will be in the form of a set of sequence S-Boxes tables arranged randomly, chosen by the two parties (sender and recipient). Figure (5-5) illustrate the process of SubByte transformation in Advanced Encryption Standard AES which using one S-box consists of 256 elements and this S-box could be generated by 8-bit value K and constant C.

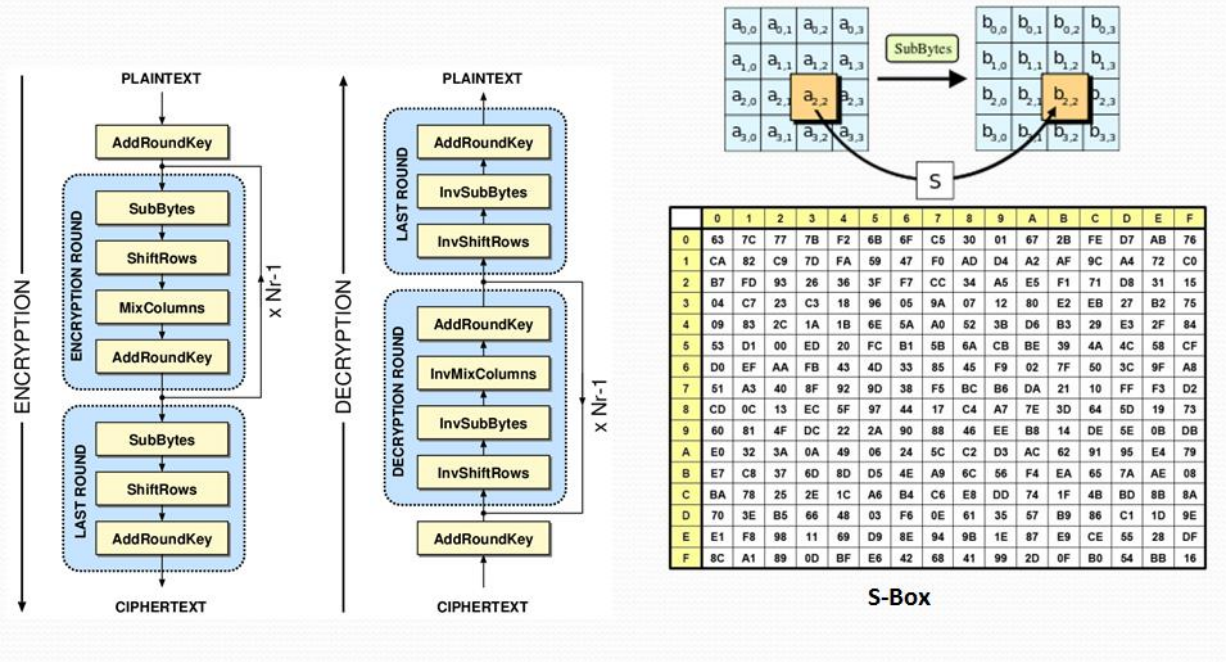


Fig. 5-5 SubByte Transformation in AES

Figure (5-6) explain the implementation of SubByte function using multi S-boxes with each S-box containing 256 numbers. Each S-box can be generated using value k and constant C. here 8 S-box are using in the encryption process

$$(S\text{-}box[m][n])k = Rndm\_Key[k] * mulp[r][c] + Cons\_c[k]$$

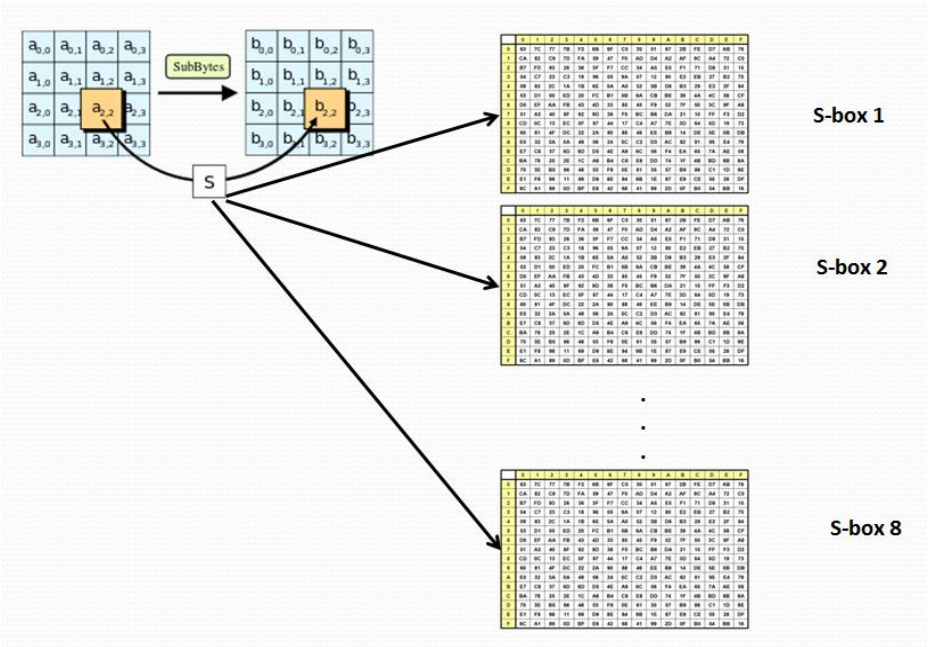


Fig. 5-6 Multi S-box Transformation

This operation leads to an increase of the degree of complexity within the same delay time during the encryption and decryption processes in the proposed SubByte function. Algorithms 1, 2 and 3 demonstrate the aforementioned processes.

Algorithm 1 S-Box generation Ali et al (2014)

***I/P**: Randomly (8) values as { Rndm_Key[k], Cons_c[k] , k=1,2,……..,8 }*
***O/P**: Diverse (8) S-Boxes { (S-box[m][n])k ; (S-1-box[m][n])k ; k=1,2,.....,8 }*
***1.** Choice 8 keys Rndm_Key[k] // to create a unique S- box when each key has the inverse to recreate Inverse S-box* ***2.** Choice Eight different values Cons_c[k]* ***3.** For each key Rndm_Key[k] & relates constant Cons_c[k] generate its own S-box[m][n]* *(S-box[m][n])k= Rndm_Key[k] * mulp[r][c] + Cons_c[k]* *Where mulp[r][c] represent the multiplicative inverse in GF(2^8)* ***4.** Use (S-box[m][n])k in encryption operation.*

Algorithm 2. Encryption with Multi S-box

***I/p** :plain text Block { State[Row][Colum] :; r,c=1,2,3,4}* *8 S-Box{(S-box[m][n])k ; (S-1-box[m][n])k ; k=1,2,.....,8 .* *matrix of Key sharing {Key-Enc\Dec[4][4]}*
***O/p:** cipher text Block {State[Row][Colum] ; r , c =1,2,3,4}*
using new S- box[m][n] to encrypt each block : ***1.** For Each Row in State matrix, Do* ***2.** For Each Colum in State matrix, Do* ***3.** Y = (Stat [Row][Colum])&0x0f;* *X = (Stat [Row][Colum]>>4)&0x0f;* *X, Y = the index of row and colum in S-Box* ***4**.State[Row][Colum] encrypt by using the index of each S-Box ,;* *Key_Enc[4][4] : State[Row][Colum]=(S-box[x][y])*

This algorithm will add much more complexity; because each key (Rndm_key and Cons_c) needed for generating the S-boxes, consist of 128 bit, this leads to a number of possible generated s-boxes potentially being:

$$complexity = 2^{128} * 2^{128} = 2^{256} \qquad (3)$$

In addition, there is a random distribution of the S-Boxes which add more complexity, by using another key led to much more complexity as:

$$random\ distribution\ \ sbox = 8!$$

The total complexity for this algorithm will, therefore, be of a high level, adding more security for this operation.

Algorithm 3. Decryption with Multi S-box

I/p *:plain text Block { State[Row][Colum] :; r,c=1,2,3,4}* *8 S-Box{(S-box[m][n])k ; (S-1-box[m][n])k ; k=1,2,.....,8 .* *matrix of Key sharing {Key-Enc\Dec[4][4]}*
O/p: *cipher text Block {State[Row][Colum] ; r , c =1,2,3,4}*
using new InvS- box[m][n] to decrypt each block : ***1.*** *For Each Row in State matrix, Do* ***2.*** *For Each Colum in State matrix, Do* ***3.*** *Y = (Stat [Row][Colum])&0x0f;* *X = (Stat [Row][Colum]>>4)&0x0f;* *X, Y = the index of row and colum in InzS-Box* ***4****.State[Row][Colum] decrypt by using the index of each InvS-Box ,;* *Key_Enc[4][4] : State[Row][Colum]=(Inv S-box[x][y])*

5.2.2 Proposed Mix S-box Operation

In This section, another technique could be proposed by using two S-boxes to increase the security level. The idea is to XOR two S-boxes and uses the new S-box in the transformation process for SubByte function. This method needs just one K and C to generate addition S-box and then XOR it with the existing one as illustrated in fig (5-7)

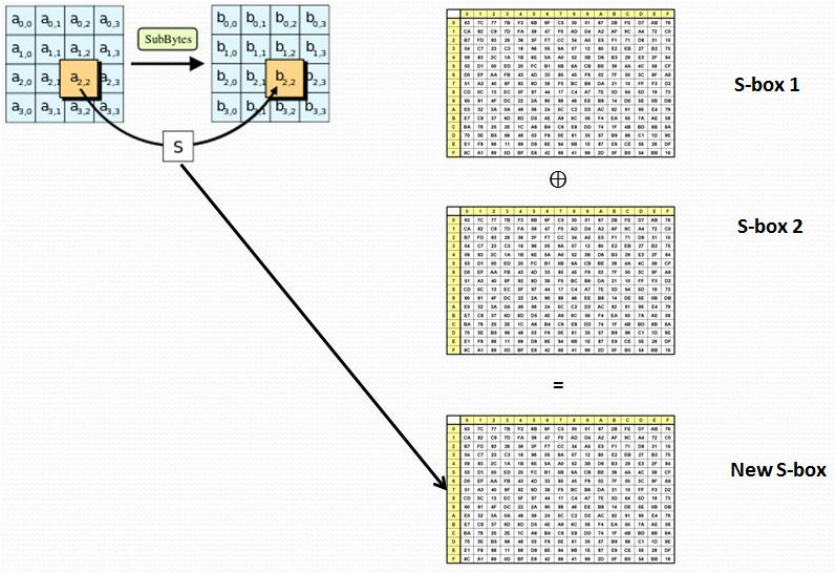


Fig. 5-7 Mix S-Box Method

This method will help to reduce the key generation size and generate just one S-box. So there is no need to generate multi S-boxes. This method can reduce the effect of using many S-boxes on the memory size of the processing unit, in addition to reduce the load on key

exchange protocol in the network function. There are 2^{16} possible S-box can be generated by the new methods using just two k (8 bit for each) as follow:

$$complexity = 2^8 * 2^8 = 2^{16}$$

This method could be used in future work and further research, so in this research, the proposed multi S-box in section (5.2.1) will be used to develop the new algorithms in this research. Algorithm 4 can explain it.

Algorithm 4 New S-Box generation with XOR operation

***I/P**: Randomly (2) values as { Rndm_Key[k], Cons_c[k] , k=1,2}*
***O/P**: S-Boxes { (S-box[m][n])k ; (S-1-box[m][n])k}*
***1.** Choice 2 keys Rndm_Key[k]* ***2.** Choice 2 values Cons_c[k]* ***3.** For each key Rndm_Key[k] & relates constant Cons_c[k] generate its own S-box[m][n]* *(S-box[m][n])k= Rndm_Key[k] * mulp[r][c] + Cons_c[k]* *Where mulp[r][c] represent the multiplicative inverse in GF(2^8)* ***4.** XOR the two (S-box[m][n])k.* *5. Get the new S-box and used in the encryption.*

5.3 Experiments

5.3.1 Testing

This test aims to execute the encryption process on the audio file using the proposed algorithm (Multi-SBOX-AES), to determine the execution time and power consumption. The testing was carried out using Visual Studio 2015 with C++ programing language. Two scenarios have been conducted in Lab's wireless laptop and computer with window 7 using a different kind of processer. In the first scenario (quad-core i7, Ram 8 GB) used. The other scenario (Due Core, Ram 4 GB) and the proposed Multi S-box algorithm have been tested on an audio file with a size of 647 KB. In the first scenario the audio file stored in the computer in drive C in the following path:

C:\Users\Fih102\Desktop\AES org\AES org

In the second scenario, the audio file stored in the following path:

\\anglia.local\fs\StudentsHome\FIH102\My Documents\Visual Studio 2015\Projects\AES org\AES org

It is very important to consider the file path and where it stores. Because this will affect the execution result, as would appear in the following result.

The program code will call the file and open and read it. Then execute the encryption process by reading and encrypting block by block, each block has 16 byte. The proposed algorithm uses dual keys; the first key is a set of multi-values up to 16 elements. Each value in the key set has another value related to it, as in AES Rijndael algorithm leading to build different S-boxes with its related inverse S-Box.

Rndom_Key[8]={**0x67, 0x85, 0x25,0xb5,0xA4, 0xf1,0x19,0x4c**}

Cons_c[8]={ **0x82, 0x45, 0xc4, 0xa5,0x7b, 0x63,0xd5,0xc1**}

Represent the first key, based on hexadecimal, each value in the key with its related cons_c value, can create unique S-Box.

The following code represents the S-boxes generation process: This code will generate eight S-boxes depending on the above keys, where each (key & con) will produce single S-box. The final algorithm implemented is:

S-boxes generation process

```
void Creat_SubByte()
{
	unsigned char r, c, value, b;
	int i;
	for (i = 0; i<8; i++)
	{
		for (r = 0; r<16; r++)
		{
			for (c = 0; c<16; c++)
			{
				value = mulp[r][c];
				b = Find_Sbox(value, Rnd_Key[i], i);
				Fill_Sbox(r, c, b, i + 1);
			}
		}
	}
```

Table 5-1 S-Box

Tables represent the S-box and its inverse which be generated by using the pairing key and correspondent constant (0x67 & 0x82).

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	82	4f	F0	DE	2f	B6	AC	06	C0	FE	98	17	01	63	54	76
1	A3	36	28	C9	1B	03	48	22	43	E8	E6	4E	7D	F1	6C	9A
2	12	8A	D8	BF	D7	65	B3	B4	DA	77	D6	A4	E7	C6	46	8C
3	62	0B	23	11	24	44	E4	6A	E9	1D	3B	47	F5	39	8E	FD
4	CA	B0	86	29	AF	2A	88	EB	BC	7F	E5	08	1A	C1	0D	21
5	3A	74	78	E2	A8	0C	05	0E	30	25	A0	72	E0	F7	85	3F
6	F2	EF	D2	9D	52	71	4B	A7	45	90	75	C4	B1	EE	F6	DF
7	37	60	D9	9E	5E	FB	F4	BE	AD	94	CB	2A	10	87	A9	FF
8	32	56	9B	64	14	C2	C3	81	80	7A	42	68	13	19	A2	EA
9	09	BD	7C	DC	A5	91	53	0F	CE	69	B7	0A	D1	92	C7	4D
A	4A	CD	F9	41	6B	6F	B2	9F	97	79	C5	04	B5	CF	50	D3
B	DB	AE	51	A1	93	6E	FA	59	27	A6	38	73	95	58	C8	4C
C	BA	55	34	8B	3E	FC	99	8D	7E	5A	7B	B5	66	2B	84	02
D	61	E3	1F	IE	ED	F3	35	5B	8F	5C	20	31	2C	1C	B8	70
E	CC	16	67	96	BB	40	18	49	EC	33	AA	F8	B9	2D	9C	57
F	15	6D	89	D0	26	5D	D4	3D	5F	E1	00	DD	83	AB	3C	07

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	FA	0C	CF	15	AB	56	07	FF	4B	90	9B	31	55	4E	57	97
1	7C	33	20	8C	84	F0	E1	0B	E6	8D	4C	14	DD	39	D3	D2
2	DA	4F	17	32	34	59	F4	D8	12	43	45	CD	BC	ED	7B	04
3	58	DB	80	E9	C2	D6	11	70	BA	3D	50	3A	FE	F7	C4	5F
4	E5	A3	8A	18	35	68	2E	3B	16	E7	A0	66	BF	9F	1B	01
5	AE	B2	64	96	E	C1	81	EF	BD	B7	C9	B7	D9	F5	74	F8
6	71	D0	30	D	83	25	CC	E2	8B	99	37	A4	1E	F1	B5	A5
7	DF	65	5B	BB	51	6A	F	29	52	A9	89	CA	92	1C	C8	49
8	88	87	00	FC	CE	5E	42	7D	46	F2	21	C3	2F	C7	3E	D8
9	69	95	9D	D4	79	BC	E3	A8	A	C6	1F	82	EE	63	73	A7
A	5A	B3	8E	10	2B	94	B9	67	54	7E	EA	FD	06	78	B1	44
B	41	6C	A6	26	27	CB	05	9A	DE	EC	C0	E4	48	91	77	23
C	08	4D	85	86	6B	AA	2D	9E	BE	13	40	7A	E0	A1	98	AD
D	F3	9C	62	AF	F6	AC	2A	24	22	72	28	B0	93	FB	03	6F
E	5C	F9	53	D1	36	4A	1A	2C	19	38	8F	47	E8	D4	6D	61
F	02	1D	60	D5	76	3C	6E	5D	EB	A2	B6	75	C5	3F	09	7F

The second key will randomly distribute the S-boxes which created by the first key. The dual keys will increase the complexity security level with roughly the same time needed for the cryptography processes. This is because instead of using single and fixed S-Box to each byte in the state matrix, the proposed algorithm uses different S-Boxes to each byte for the cipher operations.

After executing the encryption program (source.cp) by the software, the command line appears and ask to enter the main encryption key. The key used in this excrement is:

Main Key = 123456789abcdef123456789abcdef12

The length of the key is 32 char, means 16 bytes.

After entering the key and hit enter, the execution starts to encrypt the file. The choosing of the file depends on the following instruction:

```
//Opens files for reading and writing input and output.
                    Err = fopen_s(&Rfile, "test.wav", "rb");
                    erw = fopen_s(&Wfile, "AESOUTP.wav", "wb");
```

For the decryption the instruction will be as:

```
                    err = fopen_s(&Rfile, " AESOUTP.wav", "rb");
                    erw = fopen_s(&Wfile, "finalOUTP.wav", "wb");
```

So, the output file in encryption process will be the input file in decryption process. The results of the present proposed algorithm have good cryptographic strength.

Proposed Encryption algorithm

16 Byte ⟵ ***Plain block***
for *(i = 0; i<9; i++)*
NewSubBytes(); // subbytes transformation proposed in algorithm (2)
ShiftRows();
MixColumns();
AddRoundKey();
Last round;
End.

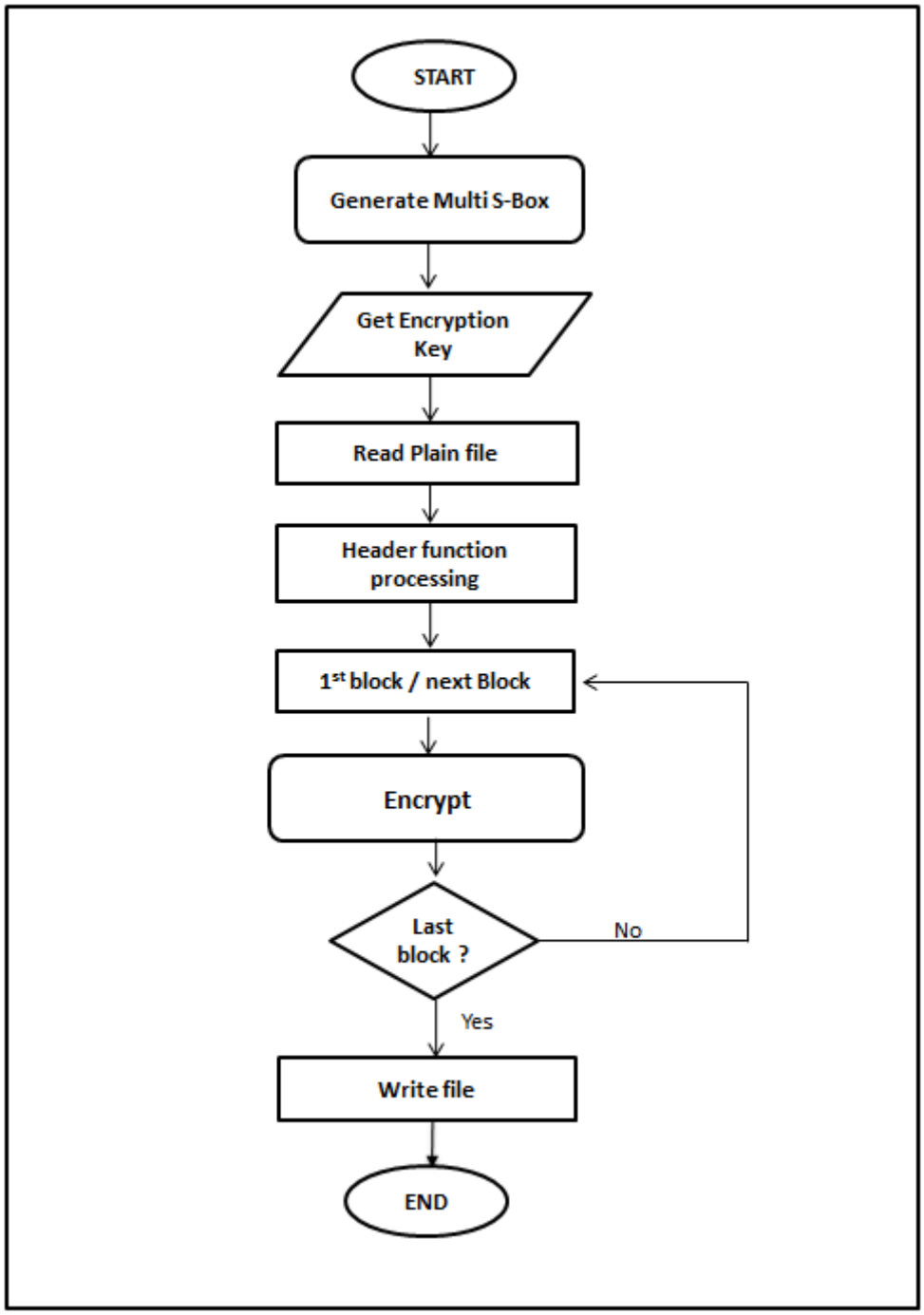


Fig. 5-8 Encryption Flow chart for Proposed Algorithm

Fig (5-9) shows the input and output files which still have the same size, and this is very important especially for networks and bandwidth.

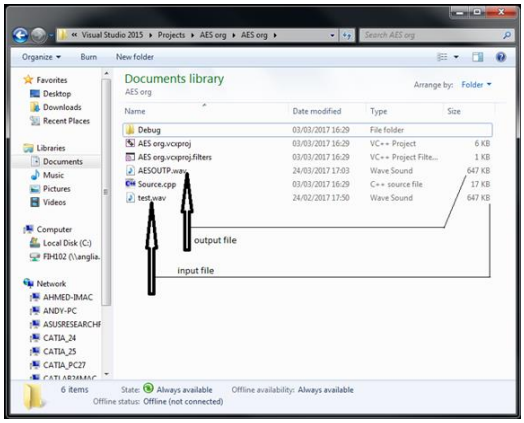


Fig. 5-9 Snapshot for output size

The result shows that the processer and the file path is very important for the execution time and the delay. In scenario 1 the delay is smaller than scenario 2 because of using (quad-core i7, Ram 8 GB) processor and storing the audio file in local folder. So our experiment shows good results with acceptable delay time. Fig (5-10) show the encryption time scenario 1 for voice file test.wav with the size of 647 KB is 1.739 sec.

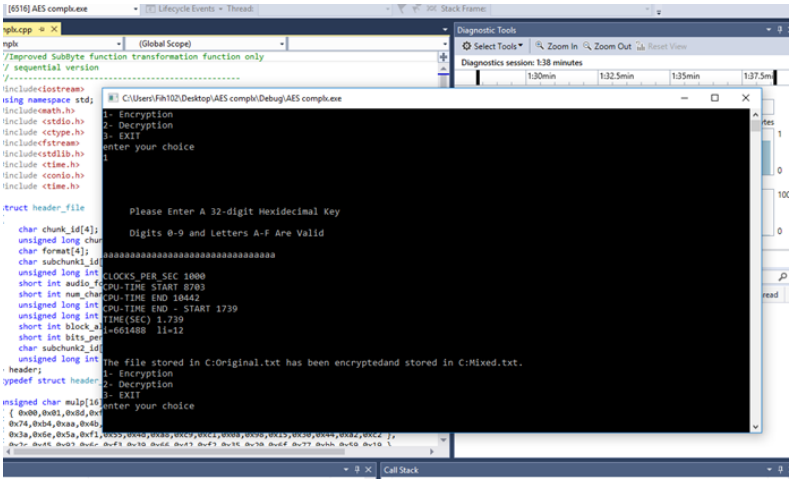


Fig. 5-10 Snapshot for run process

5.3.2 Results & Analysis

The results show that significant improvements in the AES algorithm were achieved. The following tables show the time is taken and the power consumed by the new algorithm for the encryption and decryption process.

Table 5-2 average of execution time

128K	Encry	Decry	540 K	Encry	Decry	1.48 M	Encry	Decry
	0.463	0.456		1.476	1.34		4.156	3.99
	0.462	0.455		1.46	1.37		4.154	3.99
	0.465	0.457		1.455	1.341		4.146	3.801
	0.465	0.459		1.464	1.342		4.152	4.01
	0.462	0.457		1.456	1.343		4.154	4.094
Average	0.4634	0.4568	**Average**	1.4622	1.3472	**Average**	4.1524	3.977

The above table shows the average execution time for 5 repeating execution of the program (more results tables can be found in appendix..). The following tables explain the time and power for each file.

Table 5-3 Results

(a) Execution Time

File Size	Pattern	Encryption(Sec)	Decryption(Sec)
128 K	Human voice	0.463	0.457
540 K	Human voice	1.462	1.347
1.48 M	Human voice	4.152	3.977

(b) Energy Consumed

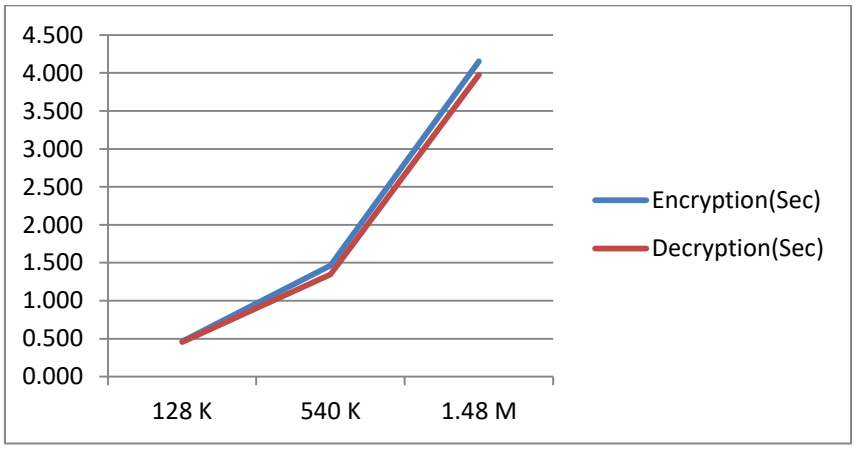
File Size	Pattern	Encr Energy(µ J)	Decr Energy(µ J)
128 K	Human voice	0.034	0.033
540 K	Human voice	0.107	0.098
1.48 M	Human voice	0.303	0.290

Table (5-3a) illustrates the amount of the execution time which has been taken by the proposed algorithm to encrypt and decrypt the audio files with different size. The lowest

amount for both the encryption and decryption processes is about 0.463 sec and 0.457 sec, respectively. While the highest 4.152 sec and 3.977 sec.

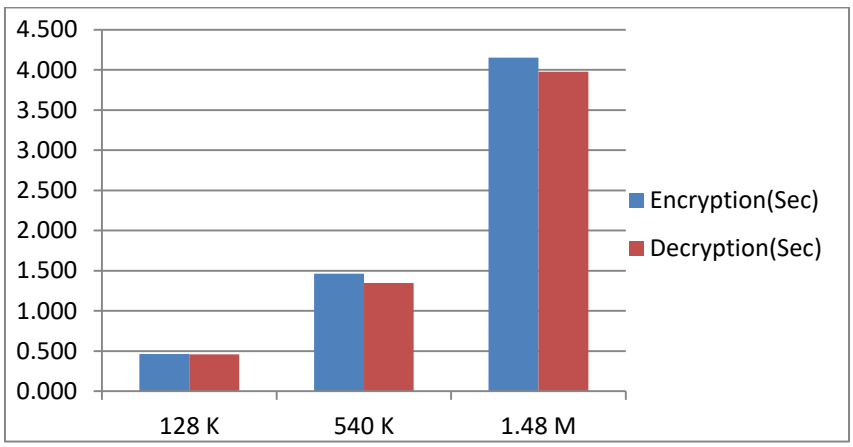
Table (5-3b) shows the power consumption which has been consumed by the proposed algorithm, it has been calculated as described in chapter three. The highest level it reached around (0.303 micro Joule) with 1.48 MB while the lowest is 0.034 (micro Joule) with 128 KB file size. The pattern of each file has been also mentioned in these tables.

The statistics figure below show the characteristic of encryption and decryption process in the proposed algorithm, it is clear that the same amount of time for both processes.



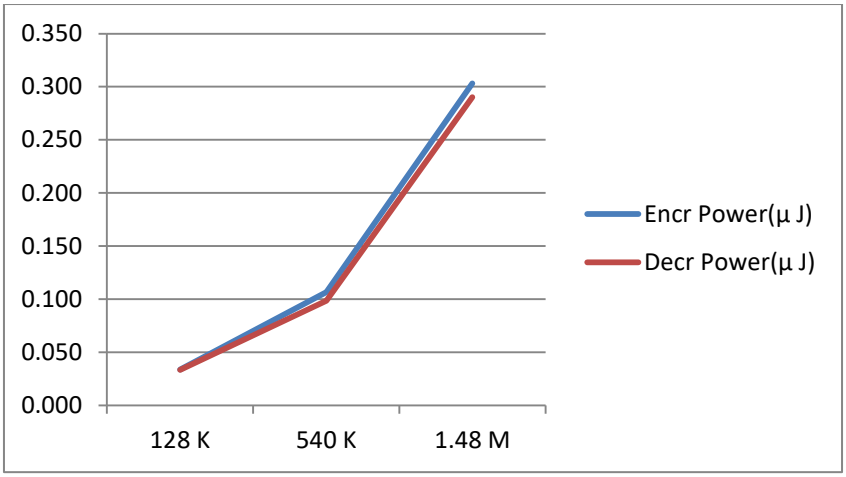
(a)

The graphs show the encryption and decryption time taken by the process in the proposed algorithm with different file sizes.

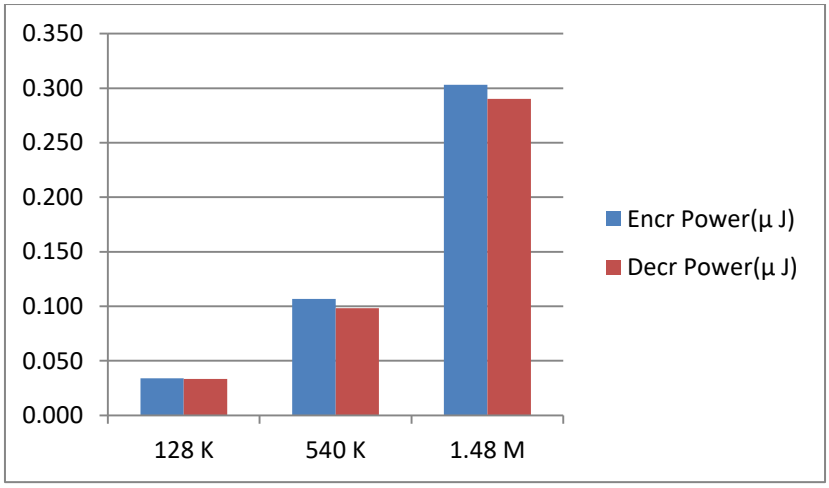


(b)

Fig. 5-11 Cryptography Time (Proposed)



(a)



(b)

Fig. 5-12 Cryptography Energy (Proposed)

5.3.2.1 Security Analysis

The new algorithm has a high-level of security because of using multi S-box in SubByte transform function. And these S-boxes are generated by using a special key and the complexity of finding the keys is increased, each keyspace is 2^{128} of complexity in addition of random distribution of the eight S_Boxes which increased the complexity by 8!, this complexity has multiplied by the number of rounds, 10 round in standard and 9 in new AES.

The figures below show the security analysis for the encrypted file for the proposed algorithm. Fig (5-13) represents the Binary Histogram of the original audio file (computer.wav) before the encryption and fig (5-14) after the encryption. It is clear that a huge difference between two figs and a good binary distribution has achieved after the encryption.

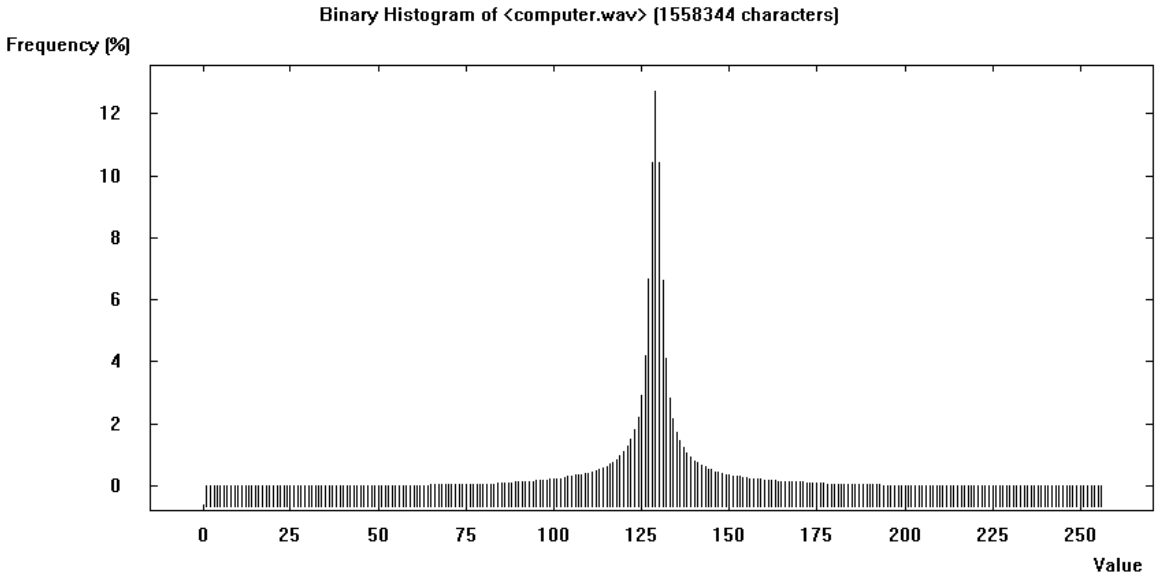


Fig. 5-13 Binary Histogram for Plain audio file

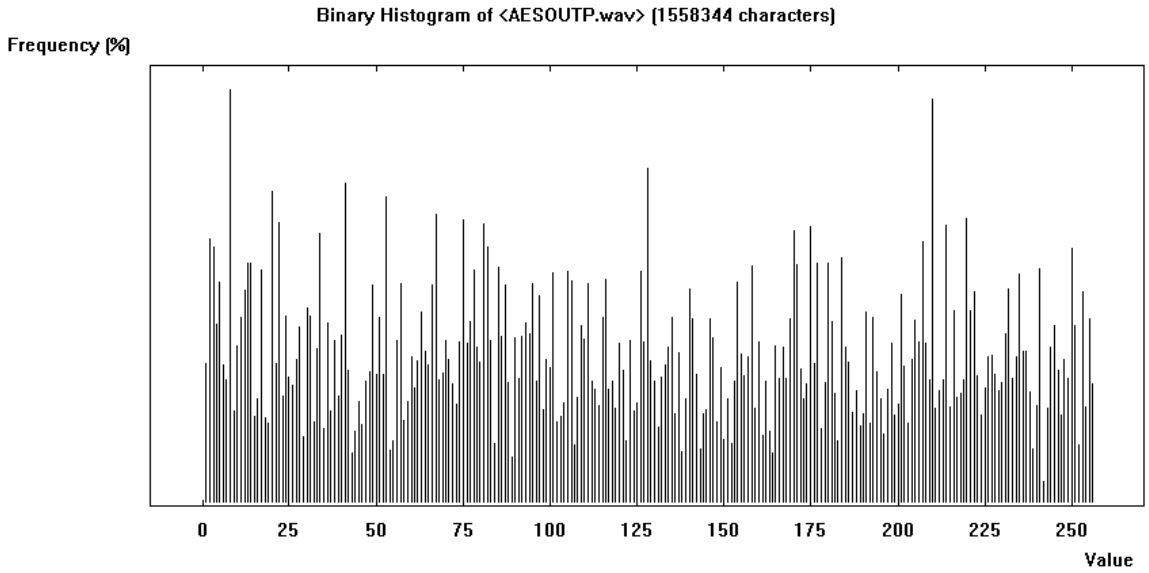


Fig. 5-14 Binary Histogram for Cipher audio file

Table 5-4 shows the Entropy test result for the encrypted file in both the AES standard and the new proposed algorithm. The new algorithm achieved a good performance compared to the standard algorithm, which was 7.99 from the maximum possible value of = 8.

Table 5-4 Entropy

Audio file	Plain	AES cipher			Proposed cipher		
	Entropy	Entropy	Max. possible Entropy	Possible byte value	Entropy	Max. possible Entropy	Possible byte value
test	7.79	7.99	8	256	7.99	8	256
teaching	5.65	7.99	8	256	7.99	8	256
washing	5.4	7.99	8	256	7.99	8	256
computer	5.13	7.99	8	256	7.99	8	256

There are no changes in the Entropy between the proposed algorithm and AES and this main that they have a similar strength.

Fig (5-15) and (5-16) shows the Floating frequency for both the original audio file and the encrypted file for the proposed algorithm.

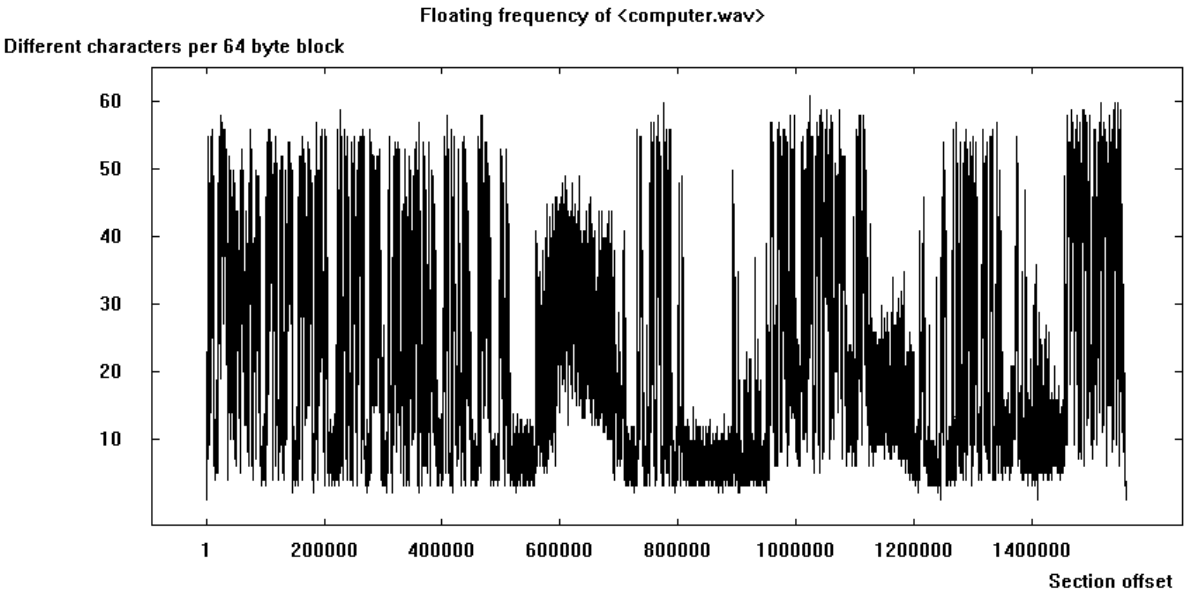


Fig. 5-15 Floating Frequency for Plain

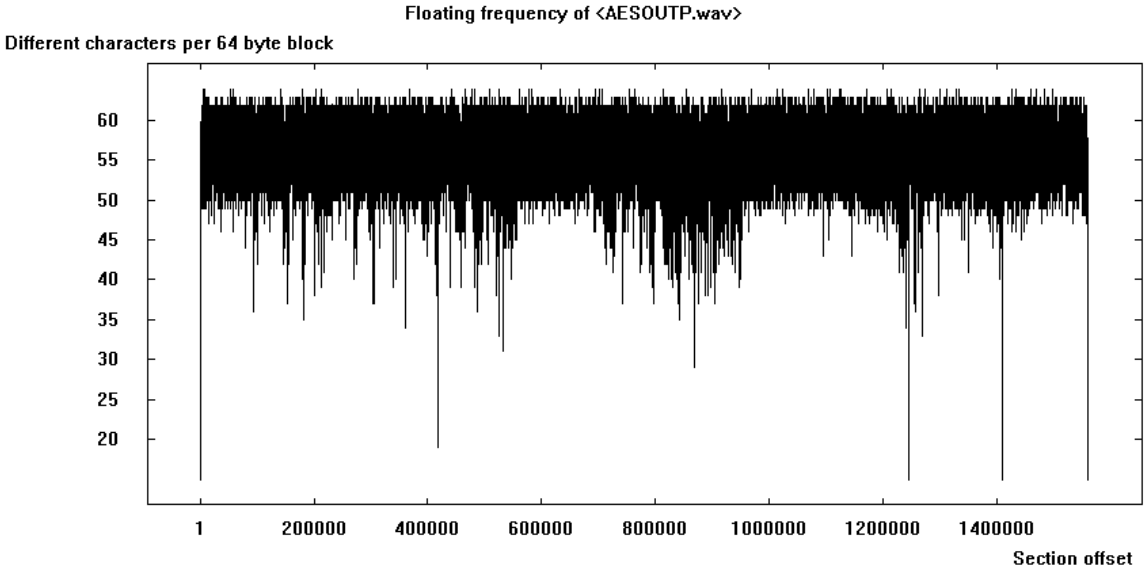


Fig. 5-16 Floating Frequency for Cipher

It is clear from the fig.5-16 that the floating frequency for the encrypted file is significant. The most of frequency is between 50 – 60 different characters per 64-byte block comparing with fig.5-15 which ranged between 10-55. This means that there is significant randomness in the newly proposed algorithm, keeping the diffusion in the cipher and leading to more complexity in the relationship between the cipher and the plaintext. All the figures above and the analysis demonstrate that an important security level has been achieved through the newly proposed encryption algorithm. In addition to the previous analysis, some tests have been carried out, such as frequency test and poker test, to test the randomness of the output audio file. These tests have been carried out on the encrypted audio file and both of these tests were passed, as shown in Fig (5-17).

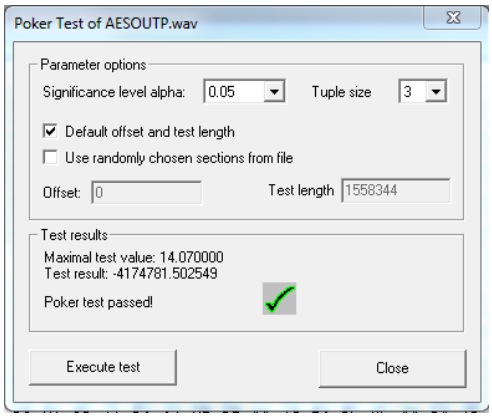


Fig. 5-17 Poker Test Result

5.3.3 Evaluation

Tables (5-5) and (5-6) show the time is taken and the power consumed by the new algorithm for the encryption and decryption process and the comparison between the traditional AES and the new encryption algorithms is provided.

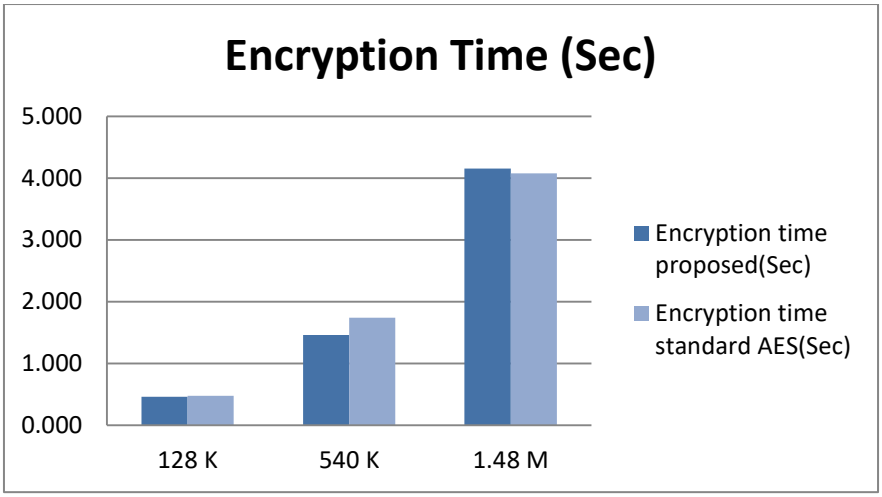
Table 5-5 AES and Proposed algorithm Time Comparison

File Size	Encryption time proposed (Sec)	Encryption time standard AES (Sec)
128 K	0.463	0.477
540 K	1.462	1.742
1.48 M	4.152	4.078

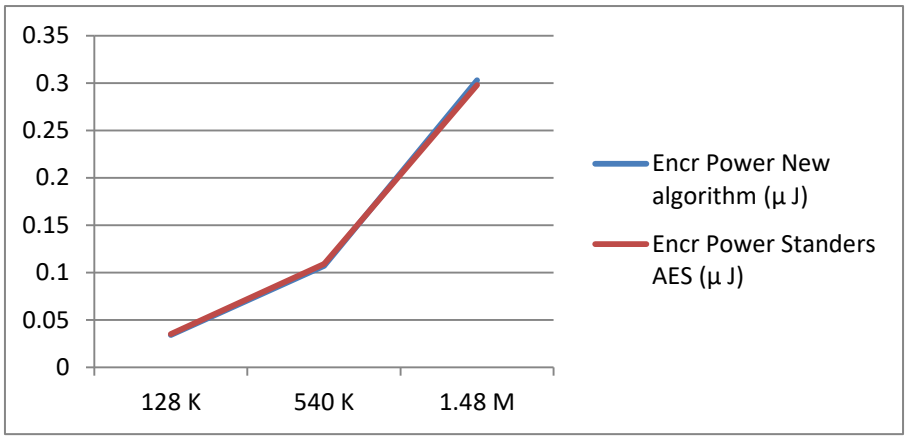
Table 5-6 AES and Proposed algorithm Energy Comparison

File Size	Encr Energy New algorithm (µ J)	Encr Energy Standers AES (µ J)
128 K	0.034	0.035
540 K	0.107	0.109
1.48 M	0.303	0.298

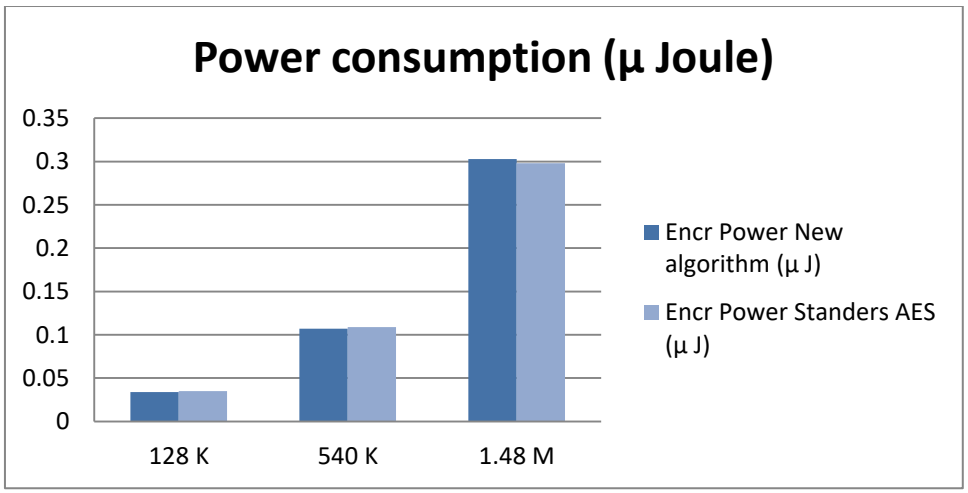
From the above table can be seen that there is the same amount in execution time and power consumption for the new algorithm. The amount still roughly the same for both the new algorithm compared with the standard AES, in spite of use multi S-boxes. The reason is because of using multi S-box does not affect the execution cost because it is a pre-processing operation. The small amount of the difference in 1.48 file is not consider because in the actual network the maximum packet size does not accede the buffer size.



(a)



(b)



(c)

Fig. 5-18 AES and Proposed algorithm comparison

Table (7-5) shows the comparison of features for both algorithms. It can be seen that there are many important differences between them. However, it is clear that the file size is still the same, which will therefore not affect the memory size and the bandwidth of network paths.

Table 5-7 A comparison between the traditional AES and the proposed algorithm

Property	Standard AES	Proposed
S_box	Single	Multi
Num. of keys	1	2
Key length	16 byte	same
Numbers of rounds	10	same
Security	2^{128}	$2^{128}*2^{128}*8!$
Single round details	Four function	same
Block length	16 Bytes	same
Encryption time	**T**	**T**
Power consumption	**P**	**P**
Output file size = input file size	Yes	Yes

This algorithm is resistant to linear and differential cryptanalysis which requires that the S-boxes be known in addition to the encryption key. From the aforementioned table, it could be noticed that the complexity is improved many times compared to the AES Rijndael, this was achieved by creating different (2 to 16) S-boxes. The structure of the AES Rijndael is the same except the creation of the S-Box and its inverse are changed. Each S-Box table is represented as 8-bit, which means the total values will be 256 and each value will appear only once in that table. In the proposed Multi S-box function, every element in the State matrix (16 elements) has a probability to map for another value of maximally $8! * 2^{256}$, since there are numerous S-boxes look-up tables generated from the first key in that function. The value 2^{256} Represent the probability appearance of every element in each S-Box and depends on multiplicative inverse used to calculate that table.

5.4 Summary

This chapter produced a primary algorithm which will be used in the next chapters to develop the new algorithms. A SubByte function using multi S-box transformation technique has been suggested to increase the confusion and complexity of encryption algorithm. The testing and experiments were conducted and a range of implementation scenarios are set up with different audio files. The complexity of the proposed algorithm has been increased and the time and energy consumption has been kept in at the same amount. Data security was analysed using specific testing tools, to measure the new algorithm strength. The following chapters will use the output of this chapter to develop the new lightweight algorithms, so many analyses, and evaluation can be found there.

5.4.1 Recommendation

Further to the proposed techniques already been presented in this chapter, there is a future possibility to propose a new function called Key XOR S-box method. The idea of this method is to XOR exist S-box with a chosen key, to increase the complexity of the algorithm. So, no need to generate multi S-boxes. The method will be also cost-effective and do not affect the execution time.

6 Chapter six: Lightweight and Low-Energy Encryption Scheme

6.1 Introduction

As explained in previous chapters, implementing current encryption method is a big issue for this kind of networks, especially with multimedia or real-time traffic, for many reasons: firstly, it needs to be more complex. And, it needs to execute quickly (Khan, et al., 2017) (Thomas & Robertazzi, 2017) (Hazzaa, et al., 2018). Also, it should be an Energy saver and do not consume a lot of battery power, as there is an Energy limitation of nodes in WNET.

AES algorithm using multi rounds iteration. The advantage of using repeated round in AES is to highly resist the cryptanalysis and to reduce the correlation in the text to achieve the diffusion (Daernen & Rijrnen, 2002). However, this consumes a lot of computer resources. So in this Research, objective #4 has been built to address these issues and reduce the Energy consumption and at the same time keeping the security at the same level.

In this chapter lightweight and low energy encryption algorithm for voice over wireless networks is being developed and tested. The new encryption algorithm has to meet the QoS requirements of voice traffic and to be suitable for wireless devices. The aim of the chapter is to reduce the execution time and energy consumption of the encryption process compared with the standard algorithm (AES) and at the same time at least maintains or increases its security level. The suggested algorithm employs similar techniques with those used in the Advanced Encryption Standard algorithm (AES), with some changes and enhancements considering the limitations of wireless devices. The test results show significant improvements in new design metrics. A range of implementation scenarios are setup; testing data is analyzed to test delay, energy, and security. Also, the comparison between the new algorithm and the standard one shows a significant amount of time and energy consumption reduction being achieved (approximately 35%), with good level of complexity, making it more suitable for the wireless environment.

Many recent types of research have been conducted to address these issues (Rahma & Yaco , 2012), (Msolli, et al., July 2016) . However, there are still some gaps that have not been addressed carefully. In (Rahma & Yaco, 2012) a Symmetric Dual key Dynamic block algorithm (SDD) for digital video in the partial encryption technology has been proposed. This algorithm meets the requirements of real-time with a high level of complexity at a considerable speed. Moreover, (Msolli, et al., July 2016) suggests a 5 rounds AES encryption

algorithm for multimedia and real-time applications in a wireless sensor network. The results showed a reduction of the execution time, however, this work is still critical in term of the security because reducing 5 round will make the algorithm very vulnerable to cryptanalysis and attacks.

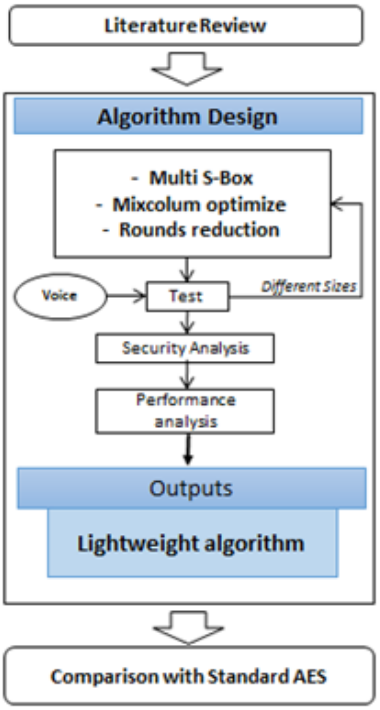
Unfortunately, all of above research results are not suitable for application with wireless networks because of the sensitive requirements it has, such as delays, throughput and power consumption in network nodes (with power limitation). The fact is that there is always a tradeoff between the QoS and the security level in encryption and network security, meaning that high security requires more processing time and consumes more energy and vice versa; these aspects have not been addressed efficiently in most network security environments. Also, there is a lack of security analysis in their work to prove the algorithms strength.

Aims

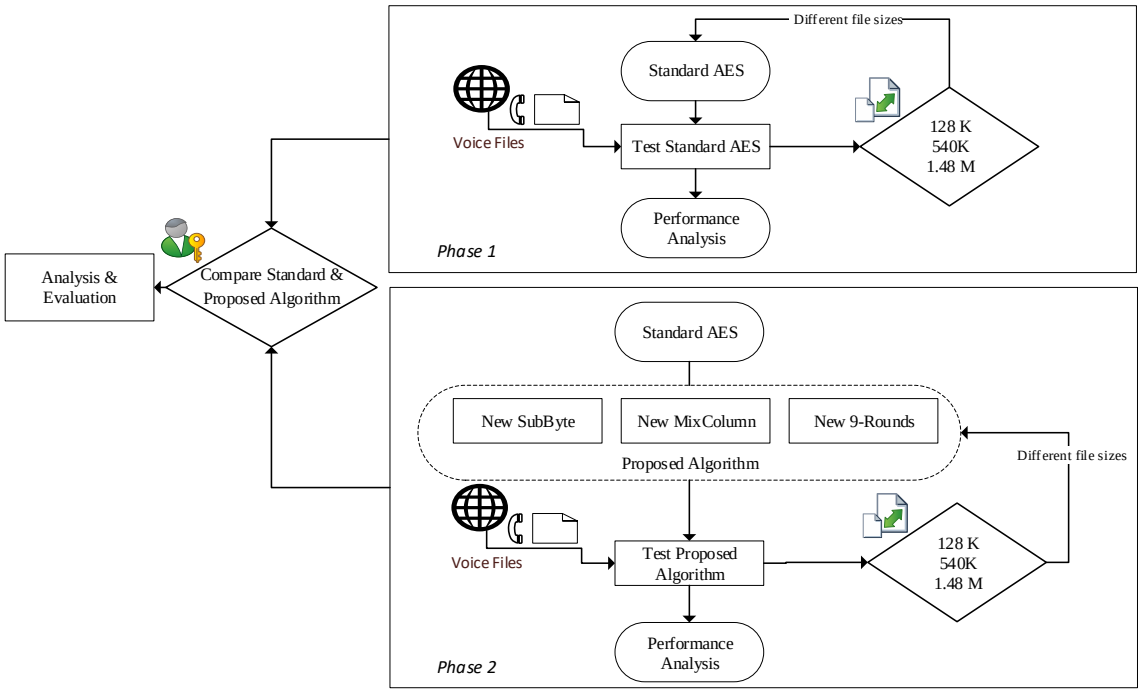
- To propose a lightweight and low-Energy encryption algorithm for voice over wireless devices.
- Using fix key for mix column function.
- Propose nine rounds encryption algorithm.
- Analyses the security parameters to prove their strength.

Methodology

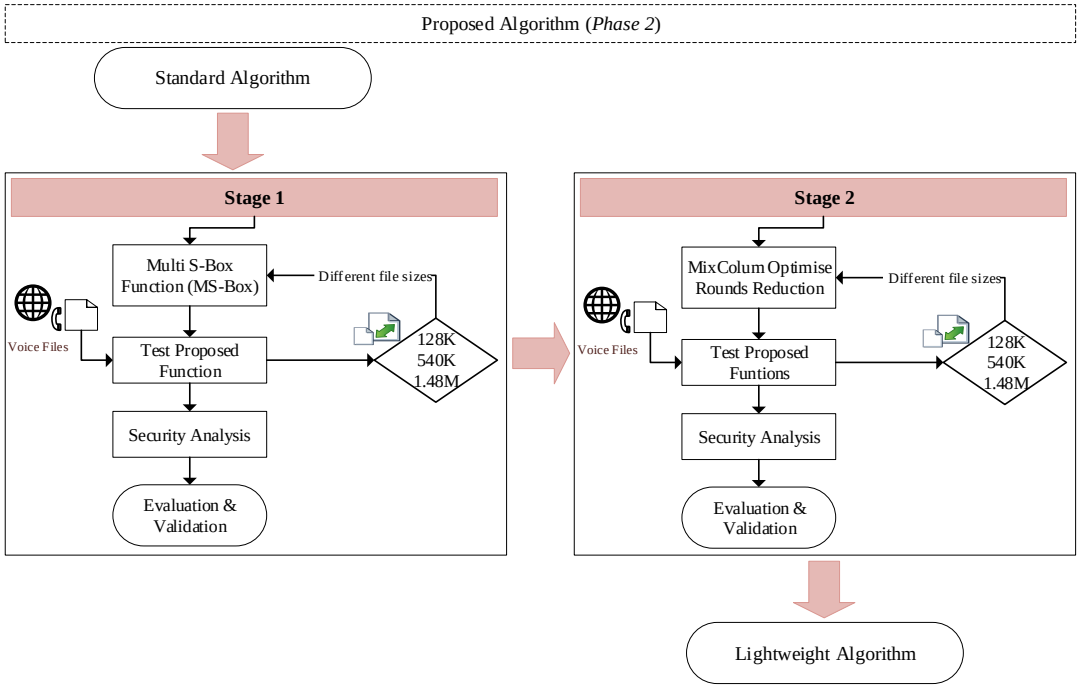
In this chapter, a quantitative research method has been adopted in which it involves running two security encryption experiments for audio files with deferent sizes. A proposed algorithm has been explained and been tested. The delay time and Energy consumed parameters have also been measured. A security analysis has been conducted to prove the algorithm strength The testing was carried out using Visual Studio 2015 with C++ programing language using Console app to avoid any load on the processor. Two scenarios have been conducted in Lab's wireless computer using two kinds of processer. In the first scenario (quad-core i7, Ram 8 GB) used. The other scenario (Due Core, Ram 4 GB) and the proposed lightweight LEA algorithm have been tested on an audio file with different sizes. Finally, the evaluation and comparison with the standard algorithm have been carried out.


Literature Review
Algorithm Design
- Multi S-Box
- Mixcolum optimize
- Rounds reduction
Voice
Test
Different Sizes
Security Analysis
Performance analysis
Outputs
Lightweight algorithm
Comparison with Standard AES

(a)


Different file sizes
Standard AES
Voice Files
Test Standard AES
128 K
540K
1.48 M
Performance Analysis
Phase 1
Analysis & Evaluation
Compare Standard & Proposed Algorithm
Standard AES
New SubByte
New MixColumn
New 9-Rounds
Proposed Algorithm
Different file sizes
Voice Files
Test Proposed Algorithm
128 K
540K
1.48 M
Performance Analysis
Phase 2

(b)



(c)

Fig. 6-1 Methodology Design

Phase 1 in fig (6-1b) has already done in chapter four. Stage 1 in phase 2 has also done in chapter five. So here in this chapter, stage 2 /phase 2 being carried out to produce the new lightweight encryption algorithm.

6.2 Proposed Development

6.2.1 Implement Low Computation Mix Column Function

The MixColumn function is the third function in the AES algorithm which represent as the most expensive operation, where the input matrix is multiplied Over $GF(2^8)$, Nada ed al (2014), (Hazzaa, et al., 2018). The key matrix used in forward and inverse MixColumn transformation functions, which operate on State matrix, is single and with fixed dimension [4*4].

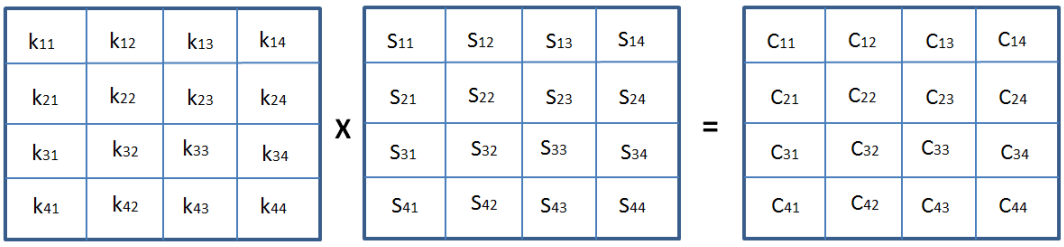


Fig. 6-2 MixColumn function in standard AES

So the operation to multiply two matrixes when [4*4] dimension for each (fig. 6-2). This means the total number of mathematic operations as follow:

$$C_{11} = (K_{11*} S_{11}) + (K_{12*} S_{21}) + (K_{13*} S_{31}) + (K_{14*} S_{41}) ,$$
$$C_{12} = (K_{11*} S_{12}) + (K_{12*} S_{22}) + (K_{13*} S_{32}) + (K_{14*} S_{42}) ,$$
$$C_{13} = (K_{11*} S_{13}) + (K_{12*} S_{23}) + (K_{13*} S_{33}) + (K_{14*} S_{43}) ,$$
$$C_{14} = (K_{11*} S_{14}) + (K_{12*} S_{24}) + (K_{13*} S_{34}) + (K_{14*} S_{44}) ,$$
$$\vdots$$
$$C_{43} = (K_{41*} S_{13}) + (K_{42*} S_{23}) + (K_{43*} S_{33}) + (K_{44*} S_{43}) ,$$
$$C_{44} = (K_{41*} S_{14}) + (K_{42*} S_{24}) + (K_{43*} S_{34}) + (K_{44*} S_{44}) \quad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

From the above equations the total number of mathematic operations can be calculated as follow: Each element in the [4*4] dimension matrix required 7 operations, two multiplications and one summation. So the total number of operations for the whole matrix is:

$$16 * 7 = 112 \text{ (mathematic operation)}$$

The proposed scheme, similar to Nada ed al (2014), tends to improve MixColumn transformation by splits the key matrix into four parts, each part representing a different key with dimension [2*2]. The State matrix is also divided into four parts with [2*2] dimension, each part corresponding to one of the keys to a similar position in the key matrix as shown in (fig.6-3).

So the operation to multiply to matrixes when [4*4] dimension for each. This means the total number of mathematic operations as follow:

$$C_{11} = (K_{11*} S_{11}) + (K_{12*} S_{21}) \quad ,$$
$$C_{12} = (K_{11*} S_{12}) + (K_{12*} S_{22}) \quad ,$$
$$C_{21} = (K_{21*} S_{11}) + (K_{22*} S_{21}) \quad ,$$
$$C_{22} = (K_{21*} S_{12}) + (K_{22*} S_{22}) \quad ,$$
$$\vdots$$
$$C_{11} = (K_{11*} S_{11}) + (K_{12*} S_{21}) \quad ,$$
$$C_{12} = (K_{11*} S_{12}) + (K_{12*} S_{22}) \quad ,$$
$$C_{21} = (K_{21*} S_{11}) + (K_{22*} S_{21}) \quad ,$$

$$C_{22} = (K_{21} * S_{12}) + (K_{22} * S_{22}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

From the above equations the total number of mathematic operation can be calculated as follow: Each element in the [4*4] dimension matrix required 3 operations, two multiplications and one summation. So the total number of operations for the whole matrix is:

$$16 * 3 = 48 \text{ (mathematic operation)}$$

In the product matrix and in each part, any element is the sum of the products of the elements of one row and one column. In this case, the individual additions and multiplications are executed in $GF(2^8)$. The transformation can be defined by the following matrix multiplication between the State and key matrices. Different from (Ali, et al., June 2014), this method assumed that all four parts of the keys are fixed. The reason is to reduce the keys exchanging operation between the sender and receiver in the network which cost more load in the network. Especially in wireless sensor networks and mobile ad hoc networks, when the nodes enter and leave frequently, this causes a repeat of rekeying operation between them.

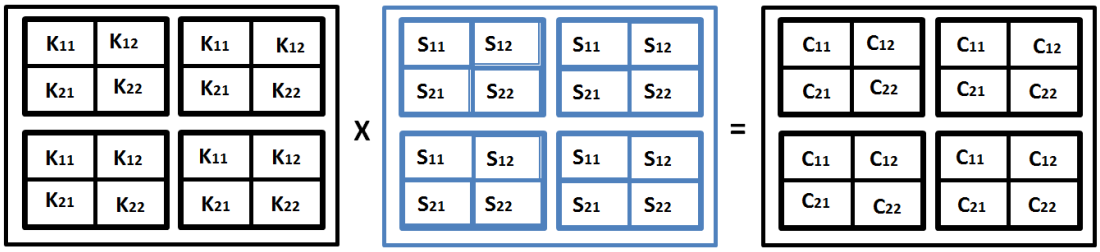


Fig. 6-3 Process of Multiplication in MixColum function with fix key

Where:

$$C11 = (K00 * S00 + K01 * S10) \; mod \; (irred.polyn.)$$

$$C12 = (K00 * S01 + K01 * S11) \; mod \; (irred.polyn.)$$

$$C21 = (K10 * S00 + K11 * S10) \; mod \; (irred.polyn.)$$

$$C22 = (K10 * S01 + \; K11 * S11) \; mod \; (irred.polyn.)$$

This method will reduce the number of operations needed to multiply two matrixes because it reduces the summation and multiplication processes. Consequently, lead to reduce the time and Energy consumption needed to executing this function, as shown in Algorithm 1.

Algorithm 1 Low Computation Mix Column

I/p :plain text Block { State[Row][Colum] :; r,c=1,2,3,4}. And The mix colum k- encryption Mix_Key[r][c], r,c = 1,……,4
O/p: Ciphertext text Block C_Blok[r][c], r,c =1,…..,4 .
1: *Divided plain Block[r][c] and Mix_Key[r][c] into 4 portions each one of size 2*2, for each portion of Mix_Key[2][2] , calculate its inverse matrix.* ***2:*** *Multiplication of each portion of P_Block[2][2]* fixed Mix_Key[2][2] to produce C_Matrix[2][2].* ***3:*** *Rebuild the C_Block[r][c] r,c=1,..,4, from each portion of C_matrix[2][2]*

6.2.2 Proposed Nine-Rounds Iteration

The high complexity that has been achieved in chapter 5 for multi S-box Sub-Byte transformation, made the algorithm much more secure. And this offers a wide range of flexibility to propose new enhancements in this algorithm, to make it suitable for wireless networks, which have nodes with power limitation. Therefore, in this section, the aim is to decrease the power consumption and execution time and keep the security and complexity at the same level.

Reducing the number of rounds in AES algorithm processes will decrease the power consumption (Hazzaa, et al., 2018) and at the same time reduces the execution time and keeps the complexity the same because the complexity was already increased in the S-Box stage as mentioned in chapter 5. It is more desirable to reduce the round iteration as much as possible. For example make it 5 rounds, as proposed by (Msolli, et al., July 2016), or 4 rounds, because it will reduce the energy consumption to the half or more. But this will breach the security level and would make the algorithm more vulnerable to the cryptanalysis. According to (Bahrak & Aref, July 2008) who confirmed that the last successful attack was on up to seven rounds AES-128. Also, the research conducted by (Li, et al., 2015) states that *"For the attacks under the single-key model, up to now, the best attacks except the biclique method could reach to 7-round for AES-128, 8-round for AES-192 and 9-round for AES-256*". So, using 8 or 9 rounds is possible, especially when increasing the complexity in other parts of the algorithm. Therefore, using 9 rounds iteration for a new algorithm is proposed. The mathematical model of the new algorithm below illustrates the proposed model:

$$P = power\ T = time$$

$$P(Enc) = \sum P(round)$$

$$T(Enc) = \sum T(round)$$

So, by implement 9 rounds:

$$New\ Power\ consumption = 0.9 * P$$

$$New\ execution\ Time = 0.9 * T$$

Implementing this new algorithm will decrease the power consumption by up to 10% and reduces the execution time to 10% as well, while keeping the security at the same level, because the complexity of the algorithm has already been dealt with during the S-Box generation stage and not in round stage, in addition, to using 9 rounds, which keep the security high (Li, et al., 2015). The AES algorithm has traditionally implemented ten rounds, each round consisting of four functions, while the proposed new algorithm is implementing 9 rounds to perform the encryption process. Algorithm 2 below illustrates the newly proposed algorithm.

Algorithm 2 Nine rounds Encryption

I/P: plaintext Block {State [Row][Col]Row, Col=1,2,3,4}. With 1-16 S-boxes, generated in ch.5.
O/P: cipher text C {[Row][Colum]Row Colum=1,..,4}

Encrypt using nine rounds
for (i = 0; i<8; i++)
SubBytes(); // subbytes transformation proposed in Ch. (5)
ShiftRows();
MixColumns(); // mixcol proposed in algorithm (1)
AddRoundKey();
Last round;
End.

6.3 Experiments

This test aims to execute the encryption process on audio files using proposed Low Computation MixColumn - 9 rounds algorithm to determine the execution time and Energy consumption. Two tests have been run. Test 1 using the proposed mixcol algorithm with 10 rounds. Test 2 used the proposed mixcol with 9 rounds.

Experiment 1 in chapter five has revealed the mixcol consumed 9/20 from the total encryption power so the theoretical mathematic would be illustrated in the following model:

the estimated total power consumption P for 10 rounds:

$$\boldsymbol{P}_{total} = (\boldsymbol{P}_{sub} + \boldsymbol{P}_{shift} + \boldsymbol{P}_{mixcol} + \boldsymbol{P}_{addkey}) * 9 + \boldsymbol{P}_{last\ round}$$

$$\boldsymbol{P}\ mixcol = \frac{9}{20} * \boldsymbol{P}\ total$$

$$new\ \boldsymbol{P}\ mixcol = \frac{1}{2} * \boldsymbol{P}\ mixcol$$

Then:

$$new\ \boldsymbol{P}\ mixcol = \frac{1}{2} * \frac{9}{20} * \boldsymbol{P}\ total$$

E= P *t

T=file size, code cost, CPU (processor speed, memory size)

There is no mixcol function in the last round so, it has not been considered in the above model. The improvement percentage should be 22% of the total power consumption; this is going to be validated in this experiment.

Also, the use of 9 rounds instead of 10 rounds in the AES algorithm should save 10% from energy. As explained in the mathematical formula.

The final decreasing percentage in power consumption should be 35% of the total power consumption.

The testing has been carried out using Visual Studio 2015 with C++ programing language. Two scenarios have been conducted in Lab's computer with window 7 using two kinds of processer. In the first scenario (quad-core i7, Ram 8 GB) is used while the other scenario is used (Due Core, Ram 4 GB). The proposed LEA algorithm has been tested on an audio file with different sizes.

The program code will call the file and open and read it. Then execute the encryption process by reading and encrypting it block by block, each block has 16 byte. The proposed algorithm uses a fixed key used as four parts; the key is a set of fixed values up to 16 elements. Each value in the key, as in AES Rijndael algorithm:

Key[16] =

{0x02,0x03,0x01,0x01,0x01,0x02,0x03,0x01,0x01,0x01,0x02,0x03,0x03,0x01,0x01,0x02}

which represent the fix key, based on hexadecimal.

The results of the present proposed algorithm have good cryptographic strength. The input and output files which still have the same size and this is very important, especially for networks and bandwidth.

The following code represents the MixColumn function:

MixColumns process

```
void MixColumns(void)
{
	unsigned char c[2][2], i, j,
		key[4][4] = { 0x02,0x03,0x01,0x01,0x01,0x02,0x03,0x01,
		0x01,0x01,0x02,0x03,0x03,0x01,0x01,0x02 },
		s[4][4], k1[2][2], m;
	for (m = 1; m <= 4; m++)
	{
		switch (m)
		{
		case 1: i = 0; j = 0;
			break;
		case 2:i = 0; j = 2;
			break;
		case 3:i = 2; j = 0;
			break;
		case 4:i = 2; j = 2;
		}
		memset(c, 0, sizeofI);  // reset the sub state matrix
		memset(k1, 0, sizeof(k1));
Cut_Arr(i, j, c, State);// take c[2X2] from each 4X4 state matrix
Cut_Arr(i, j, k1, key);
		Product(c, k1);
Re_const(c, s, i, j);
	}
	for (i = 0; i<4; i++)
		for (j = 0; j<4; j++)
			State[i][j] = s[i][j];
```

The code will execute the new mix-column process by dividing the 16-byte state to four parts with 2 dimensions for each and doing the multiplication operation.

After executing the encryption program (source.cp) by the software, the command line appears and asks to enter the main encryption key. The key used in this excrement is:

Main Key = 123456789abcdef123456789abcdef12

The length of the key is 32 char, means 16 bytes, 128 bit. Also, the S-box key generation will be used here, as explained in chapter 5 to use multi S-Box.

After entering the key and hit enter, the execution starts to encrypt the file. The choosing of the file depends on the following instruction:

```
	//Opens files for reading and writing input and output.
		Err = fopen_s(&Rfile, "test.wav", "rb");
		erw = fopen_s(&Wfile, "AESOUTP.wav", "wb");
```

For the decryption the instruction will be as:

```
err = fopen_s(&Rfile, " AESOUTP.wav", "rb");
erw = fopen_s(&Wfile, "finalOUTP.wav", "wb");
```

So, the output file in encryption process will be the input file in decryption process. The encryption key should be between 0-9 and A-F hex character. Otherwise, it shows error alert, as shown in fig (6-4).

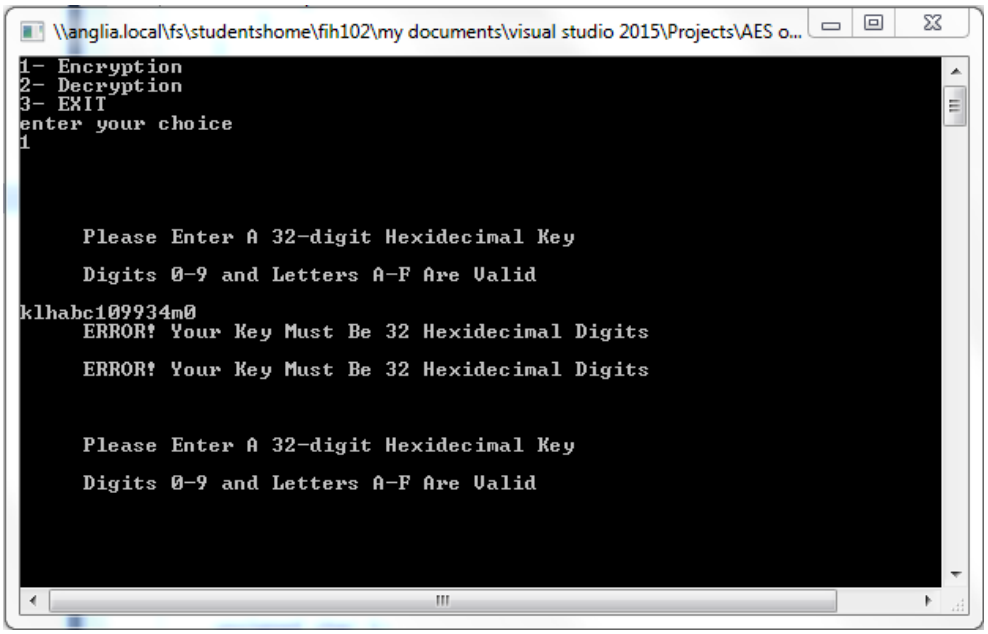


Fig. 6-4 Snap shot for Key error

6.3.1 Results & Analysis

6.3.1.1 Low Computation Mix Column Results

Tables (6-1 a, b) show the time is taken and the energy consumed by the new low computation mixcolumn algorithm for the encryption and decryption process for audio files. It describes the test result by using (quad-core i7, Ram 8 GB) processor:

Table 6-1

(a) Testing Time low computation

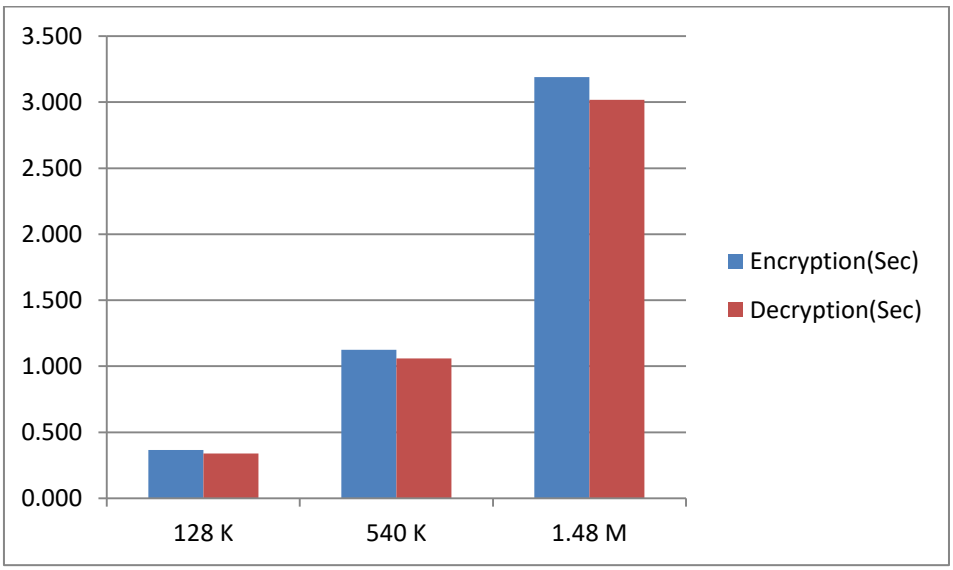
File Size	Encryption Time (Sec)	Decryption Time (Sec)
128 K	0.365	0.339
540 K	1.125	1.060
1.48 M	3.190	3.017

(b) Testing Energy

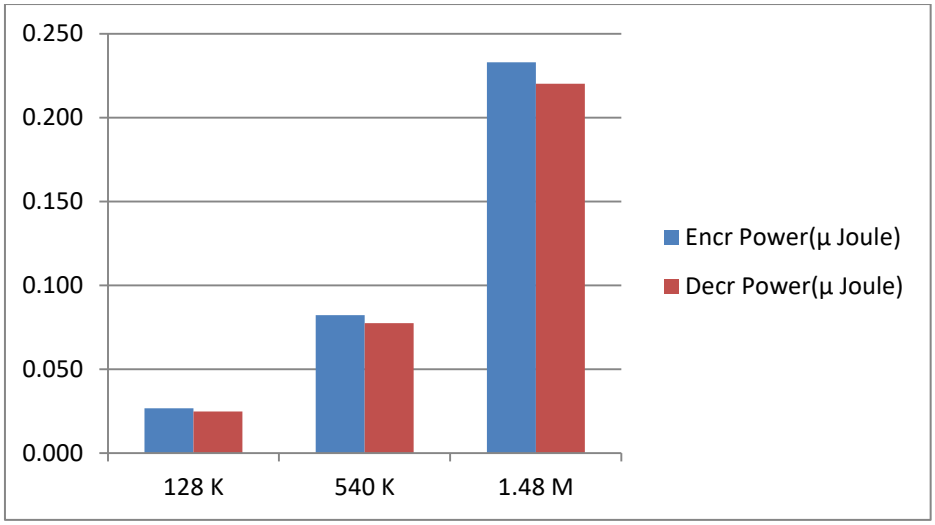
File Size	Encryption Energy (µ Joule)	Decryption Energy (µ Joule)
128 K	0.027	0.025
540 K	0.082	0.077
1.48 M	0.233	0.220

Tables (6-1a, b) illustrate the amount of the execution time which has been taken by the proposed lightweight algorithm to encrypt and decrypt the plain audio files with different sizes. The highest level it reached nearly 3.190 sec with 1.4 MB while the lowest is 0.365 sec with 128 KB file size. Table 2 shows the energy consumption which has been consumed by the proposed algorithm, it has been calculated as described in chapter three. The highest level it reached approximately 0.233 µ Joule while the lowest is 0.027 µ Joule with 128 KB file size.

The following graphs show the encryption and decryption time and energy taken by the processes in the proposed low computation mixcolumn algorithm with different file sizes.



(a)



(b)

Fig. 6-5 Encryption\Decryption Time and Energy low computation

6.3.1.2 Lightweight algorithm Nine-Rounds Results

Tables (6-2) show the time is taken and the energy consumed by the new lightweight LEA algorithm for the encryption and decryption process.

Table 6-2 (a) Testing Time lightweight LEA

File Size	Encryption Time (Sec)	Decryption Time (Sec)
128 K	0.314	0.324
540 K	0.977	1.086
1.48 M	2.747	2.833

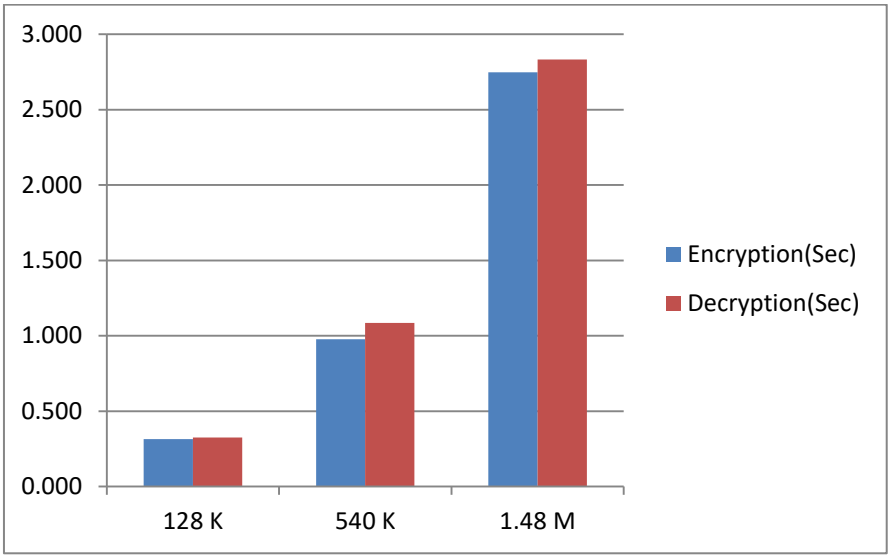
(b) Testing Energy

File Size	Encryption Energy (µ Joule)	Decryption Energy (µ Joule)
128 K	0.023	0.024
540 K	0.071	0.079
1.48 M	0.201	0.207

Table (6-2 a, b) illustrates the amount of the execution time which has been taken by the proposed lightweight algorithm to encrypt and decrypt the plain audio files with different sizes. The highest level it reached nearly 2.747 sec with 1.4 MB while the lowest is 0.314 sec

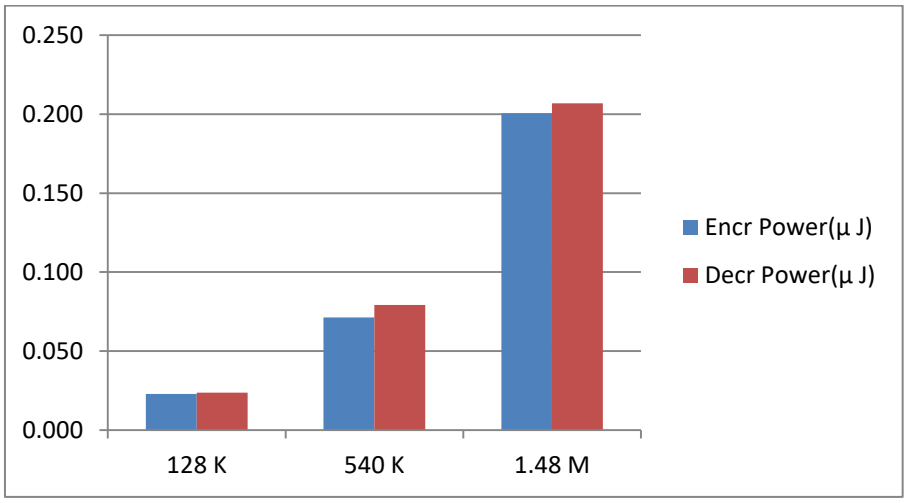
with 128 KB file size. Table 2 shows the energy consumption which has been consumed by the proposed algorithm, it has been calculated as described in chapter three. The highest level it reached approximately 0.201 µ Joule while the lowest is 0.023 µ Joule with 128 KB file size.

The statistics figure below show the characteristic of encryption and decryption process in the proposed algorithm, it is clear that the same amount of time for both processes.



(a)

Fig. 6-6 Encryption\Decryption Time and Energy lightweight LEA



(b)

The following table shows the execution time and Energy consumption for LEA run with different encryption s' key. It is clear that the changing of the key does not affect the encryption\ decryption processing time and energy, as compared with a table (6-2).

Table 6-3 Different Encryption key

	Execution Time		Energy Consumption	
File Size (B)	**Encryption Time (Sec)**	**Decryption Time (Sec)**	**Encryption Energy (µ J)**	**Decryption Energy (µ J)**
128 K	0.314	0.324	0.023	0.024
540 K	0.974	1.086	0.071	0.079
1.48 M	2.745	2.8	0.2	0.205

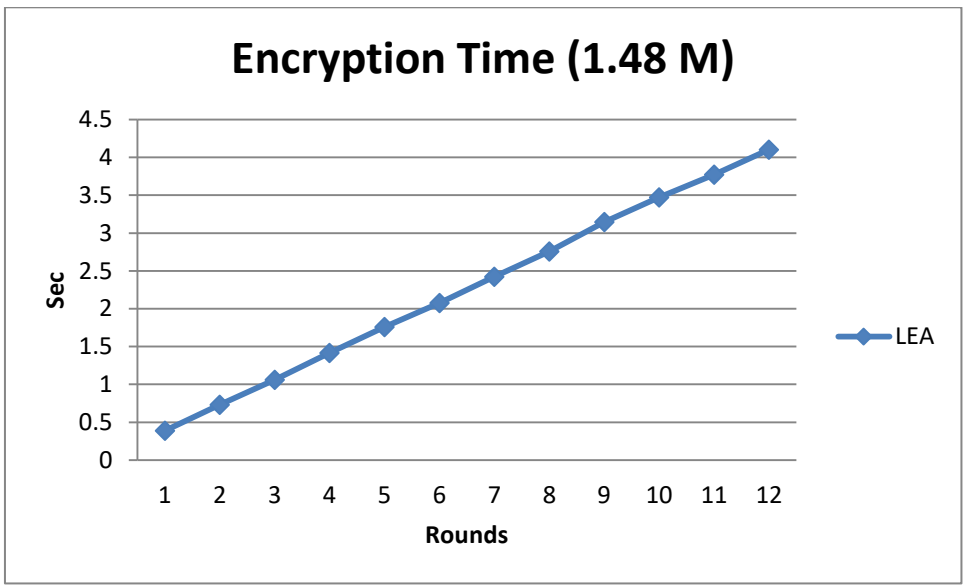
The experiment has also tested the new proposed algorithm with many numbers of encryption rounds to show the differences between them and their effect on the encryption cost. The following tables illustrate the results for each round.

Table 6-4 Execution Time with many rounds

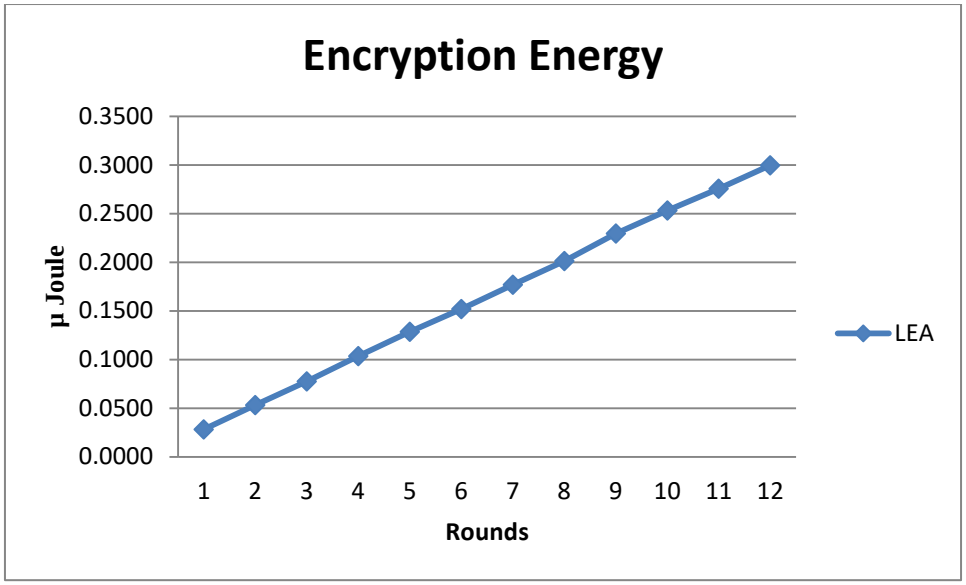
Rounds Iteration	**Encryption Time (Sec)**	**Decryption Energy (µ J)**
1st Rn	0.387	0.387
2nd Rn	0.73	0.73
3rd Rn	1.061	1.09
4th	1.416	1.421
5th	1.759	1.76
6th	2.078	2.078
7th	2.424	2.424
8th	2.755	2.757
9th	3.144	3.145
10th	3.47	3.47
11th	3.773	3.77
12th	4.104	4.1

Table 6-5 Energy consumption with many rounds

Rounds Iteration	Encryption Energy (µ J)	Decryption Energy (µ J)
1st Rn	0.0283	0.0283
2nd Rn	0.0533	0.0533
3rd Rn	0.0775	0.0796
4th	0.1034	0.1037
5th	0.1284	0.1285
6th	0.1517	0.1517
7th	0.1770	0.1770
8th	0.2011	0.2013
9th	0.2295	0.2296
10th	0.2533	0.2533
11th	0.2754	0.2752
12th	0.2996	0.2993



(a)



(b)

Fig. 6-7 Execution Time & Energy for LEA with many rounds

6.3.1.3 Security Analysis

Appropriate security parameters are required to investigate the degree of randomness and encryption quality for the output file (binary sequences) produced by the proposed algorithm, statistical testing and mathematical measurements (Riad, et al., 2013). And then these metrics could be used to collect evidence whose output sequences are truly random and have a high encryption quality, which can be used safely in the converged network applications (Rana & Wankhade, 2017; Jonge & Loo, 2013).

The security analysis has given further consideration in this chapter to confirm the security strength of the novel design. There are two proves to confirm the security of the encrypted files: the human recognition of the recorded sound, the CrypTool diagrams, and results. The recorded sound (cipher) after encryption was completely unclear and nobody can understand it. However, there are many security tests conducted in this section.

The figures below show the security analysis for the encrypted file for both the standard AES and the proposed algorithm. Fig (6-8) represents the Binary Histogram (as explained in ch.3) of the original audio file (test.wav) before the encryption.

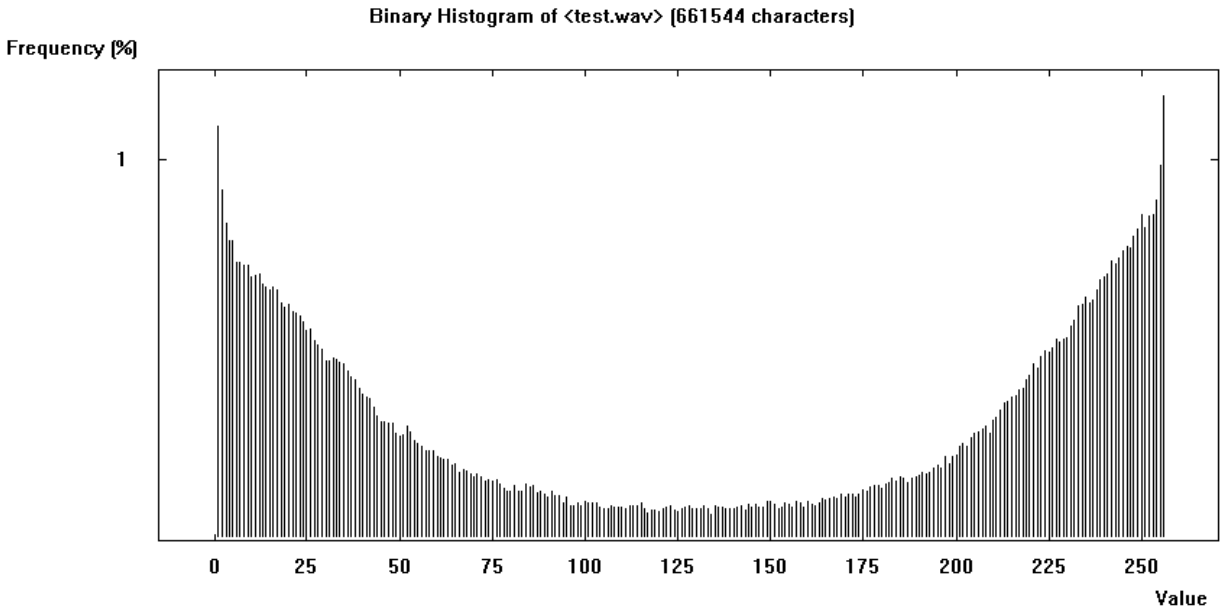


Fig. 6-8 Binary Histogram for Plain audio file

Fig (6-9) illustrates the Binary Histogram of the encrypted file by the proposed algorithm. It is clear that there is a huge difference between the original file and the encrypted file. The above figures describe the repetition (distribution) percentage of each character/number in the audio file. For example, number 50 has a different value in each figure, this means that the encrypted file has good confusion pattern to trick the attacker keeping the data more secure.

Fig (6-10) shows the Binary Histogram of the encrypted file for the standard AES. As a comparison with the Binary Histogram presented for the new algorithm in Fig (6-9), there is a clear similarity between the standard AES and the proposed algorithm as compared with the original file.

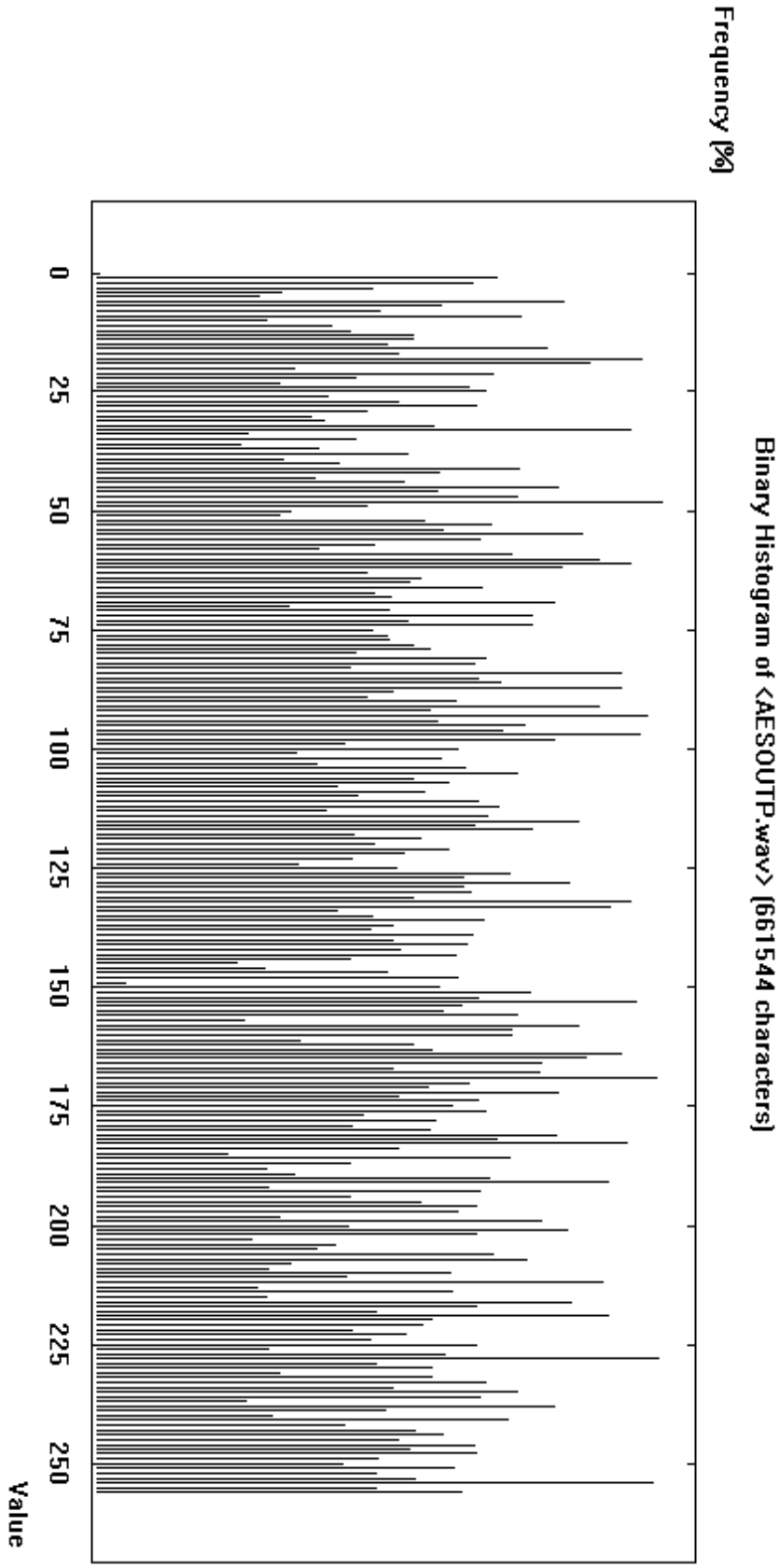


Fig. 6-9 Binary Histogram for Cipher audio file by LEA

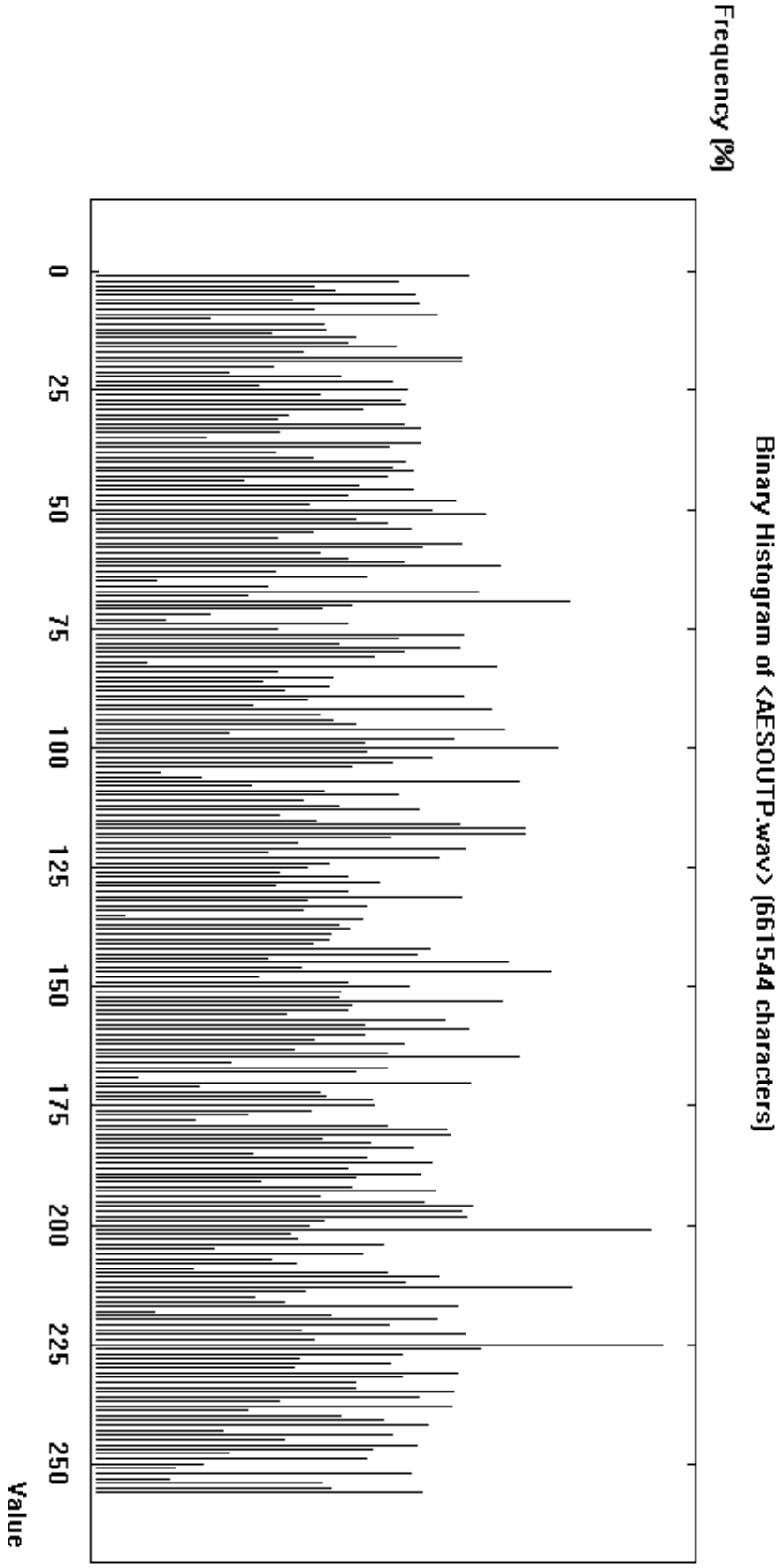


Fig. 6-10 Binary Histogram for Cipher audio file by AES

The following table describes the previous diagrams and illustrate their values. It shows the frequency for some characters for the plain and cipher which encrypted by standard AES and the proposed Lightweight Encryption Algorithm (LEA).

Table 6-6 Binary Histogram Values

Character value (Dec.)	Equivalent value (Hex)	Frequency (Plain Text) (%)	Frequency Cipher (LEA) (%)	Frequency Cipher (AES) (%)
1	01	**1**	0.68	0.61
25	19	**0.49**	0.67	0.51
50	32	**0.25**	0.33	0.55
75	4B	**0.13**	0.47	0.31
100	64	**0.09**	0.62	0.75
125	7D	**0.06**	0.50	0.35
126	7E	**0.061**	0.70	0.31
127	7F	**0.062**	0.61	0.41
128	80	**0.063**	0.82	0.46
150	96	**0.12**	0.57	0.52
175	AF	**0.13**	0.61	0.47
200	C8	**0.22**	0.42	0.37
225	E1	**0.46**	0.62	0.93
250	FA	**0.77**	0.40	0.18

Frequency means the number of times repeated in the text. From the above table it is clear that a good confusion has been achieved in the cipher for both TKE and AES, For example, the character (125) in the diagram, (125 Dec.= 7D Hex= } ASCII), has 0.5% frequency, While actual frequency in the plain file is 0.06%. Also, 75 has 0.47% While actual frequency in the plain file is 0.13%. So, all the frequency values have big differences from the plain file. This adding more confusion to the cipher and will confuse the attacker making it more secure and high resistance to cryptanalysis. To calculate the differences between the Plain file and cipher mathematically, the following equation computes the number of each character in the file:

$$no.of\ char = Total\ no.of\ char * \frac{char\ freq.}{100}$$

For example, the number of character repeating (125≠7D) =

$$no.of\ 7D = 661544 * \frac{0.5}{100} = 3307$$

While in the Plain file:

$$no.of\ 7D = 661544 * \frac{0.06}{100} = 397$$

Another example of (150≠96):

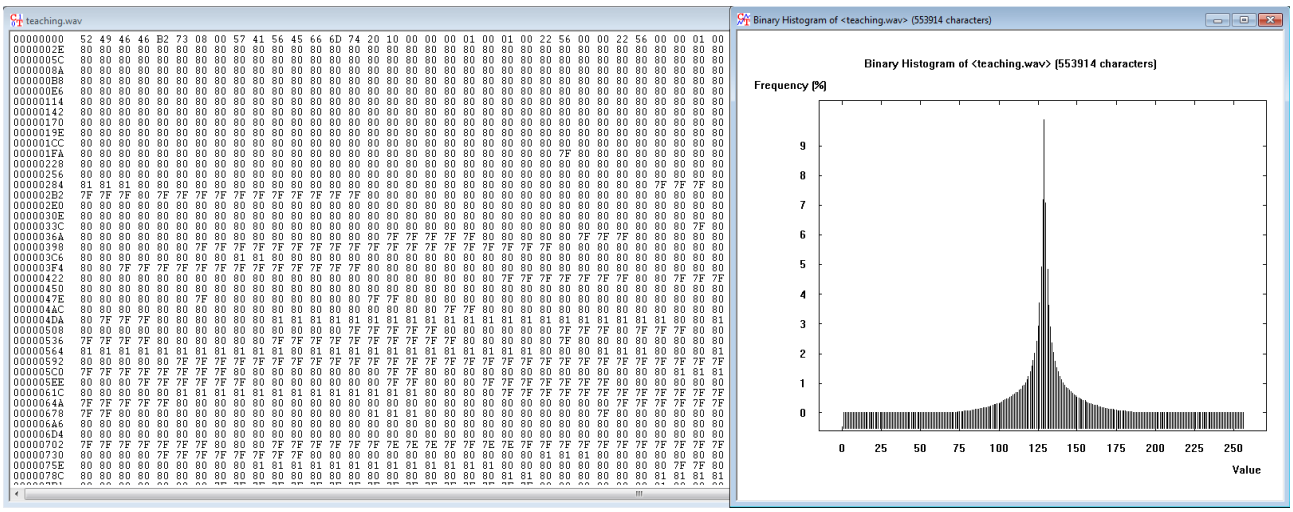
$$no.of\ 96 = 661544 * \frac{0.57}{100} = 3770$$

In Plain file:

$$no.of\ 96 = 661544 * \frac{0.12}{100} = 794$$

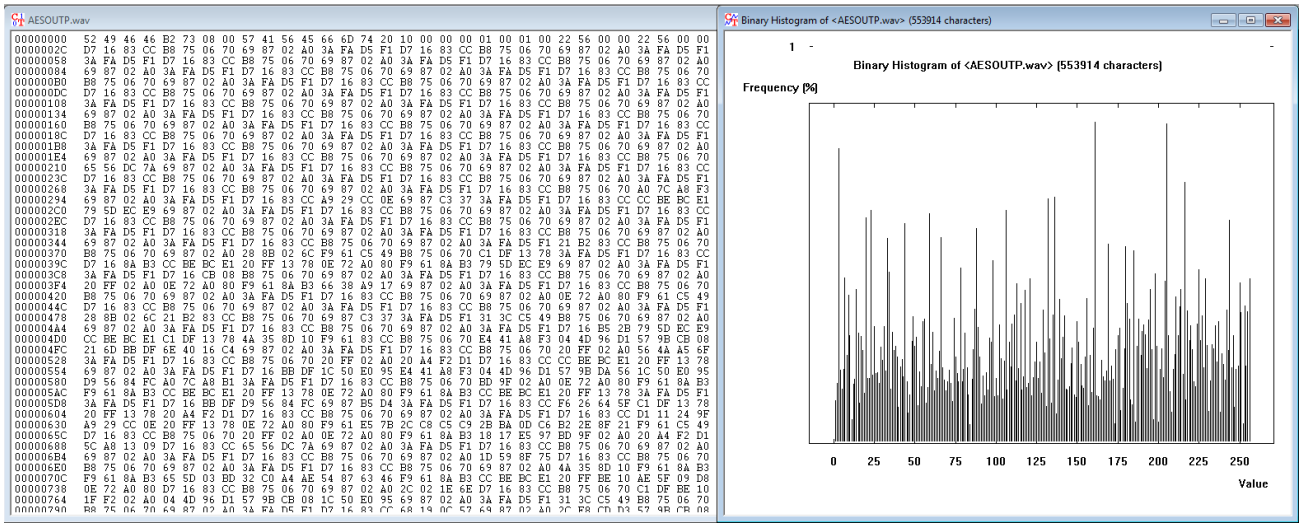
Thus, by using the proposed cryptosystem, the cipher will not supply any useful information related to the plain-file. And because there are significant differences between the plain and cipher file, the statistical attacks will be infeasible.

Additional audio file (teaching.wav 540k) has also been considered in this test and the following figs show their pattern.

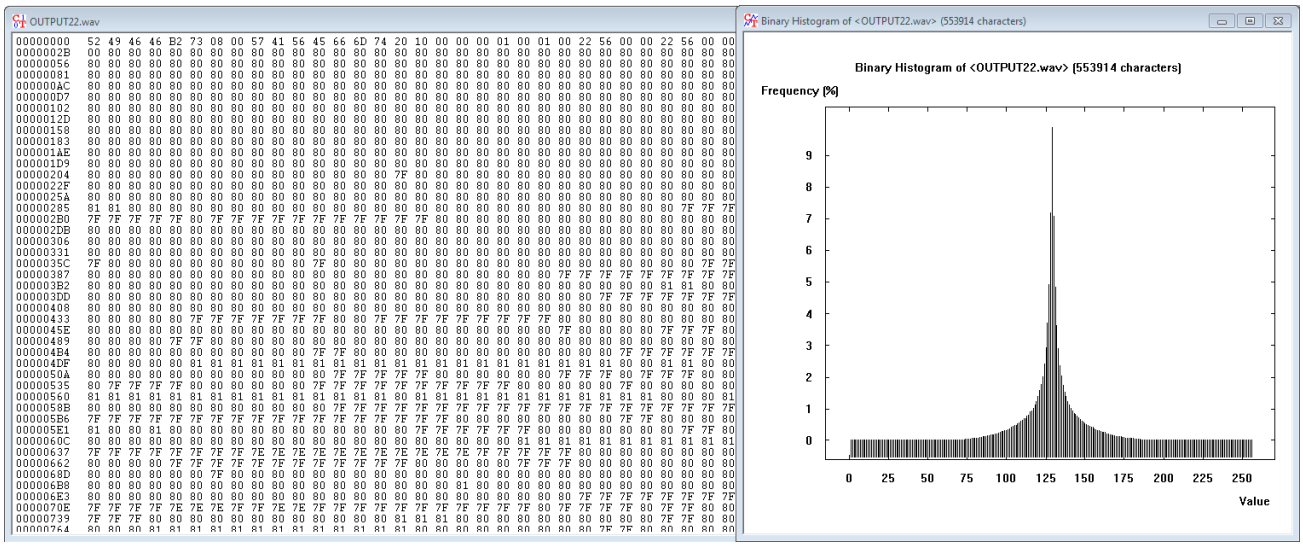


(a) Plain file

Fig. 6-11 Hex file and binary histogram for teaching file

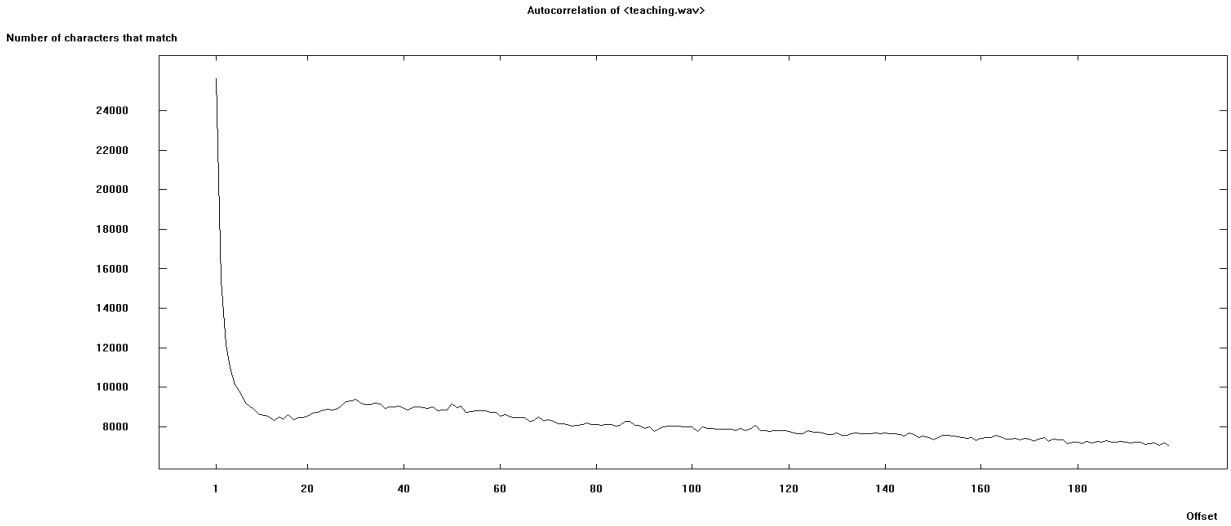


(b) Cipher

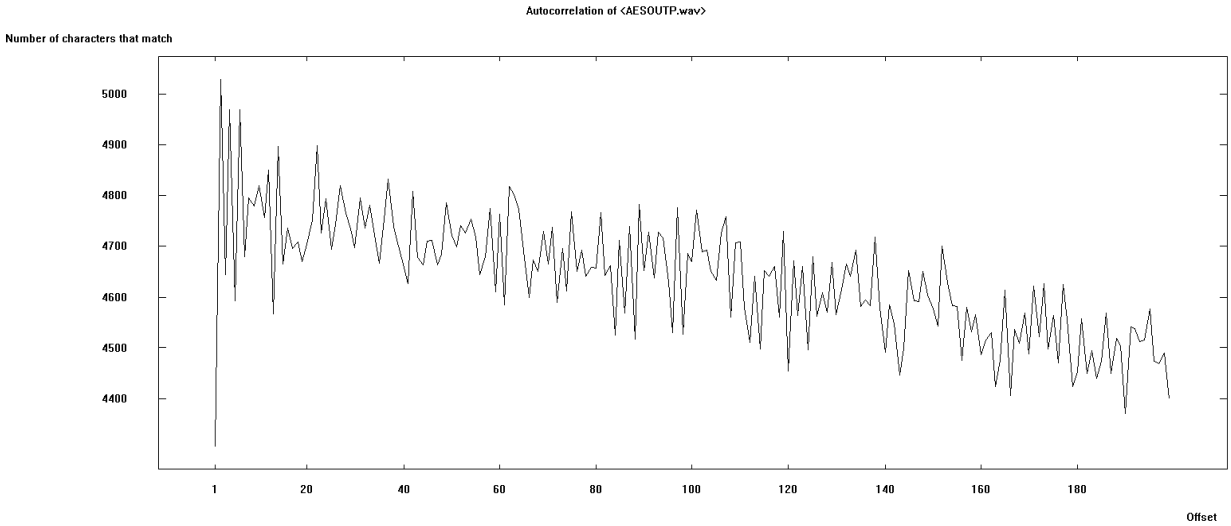


(c) Decipher

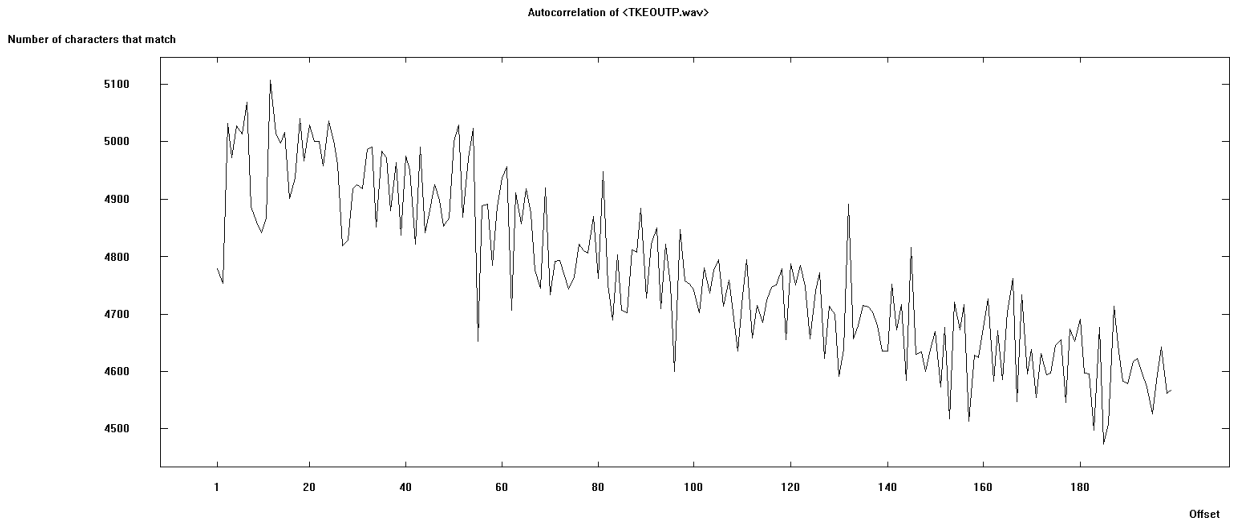
Another security parameter called Autocorrelation, (as explained in ch.3), has been considered in this analysis to show the correlation pattern for the plain and cipher. The following graphs illustrate its analysis.



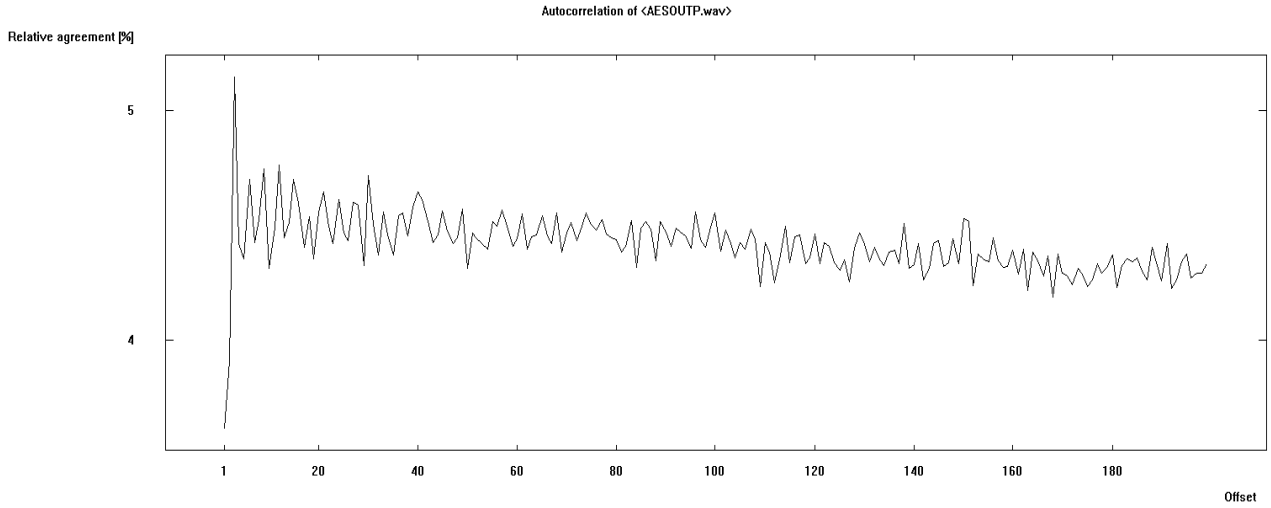
(a) Autocorrelation for Plain file



(b) Autocorrelation for Cipher by AES



(c)Autocorrelation for Cipher by TKE



(d) Relative agreement % for Cipher

Fig. 6-12 Autocorrelation for testing audio file

Table (6-7) describe the previous autocorrelation figs and illustrate the numerical values for each test. There is a big reduction in the correlation in the ciphers.

Table 6-7 Autocorrelation

Audio file	Plain		LEA		AES		Degree
	no. match	% relative agreement	no. match	relative agreement %	no. match	% relative agreement	
teaching	8000-11000	20%	4700-5200	4%	4500-5000	4%	P
Washing	48000-55000	12-15%	8200-8600	4%	8400-8800	4%	P
computer	19000-40000	15%	12400-13400	5%	12400-13400	5%	P

The number of characters that match is reduced in the cipher for LEA and AES and it roughly the same amount. This means that have been a similar cryptographic strength for both of them, according to ch.3.

Table (6-8) shows the Entropy test result for the encrypted file in both the AES standard and the new proposed algorithm. The new algorithm achieved a good performance compared to the standard algorithm, which was 7.99 from the maximum possible value of = 8.

Table 6-8 Entropy

Audio file	Plain	AES cipher			LEA cipher		
	Entropy	Entropy	Max. possible Entropy	Possible byte value	Entropy	Max. possible Entropy	Possible byte value
test	7.79	7.99	8	256	7.99	8	256
teaching	5.65	7.99	8	256	7.99	8	256
washing	5.4	7.99	8	256	7.99	8	256
computer	5.13	7.99	8	256	7.99	8	256

The floating frequency describes the number of different characters per 64-byte block, higher number means higher security (Riad, et al., 2013). Fig (6-13) and (6-14) shows the Floating frequency for both the original audio file and the encrypted file for the proposed algorithm.

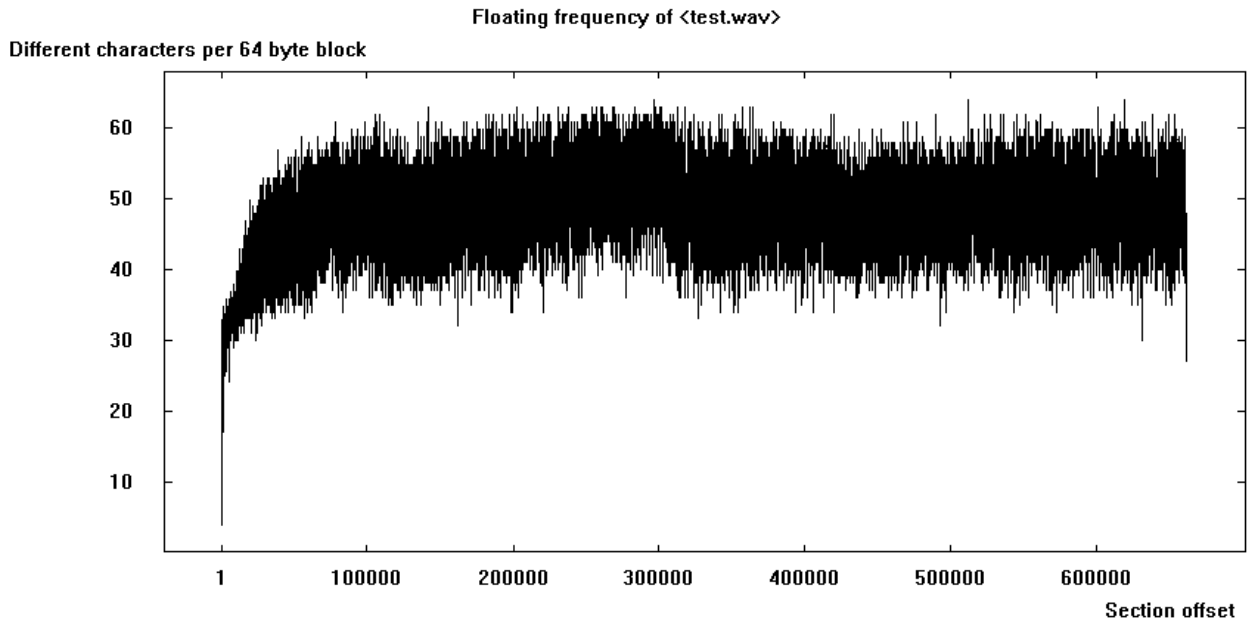


Fig. 6-13 Floating Frequency for Plain file

This means that there is significant randomness in the newly proposed algorithm, keeping the diffusion in the cipher and leading to more complexity in the relationship between the cipher and the plaintext.

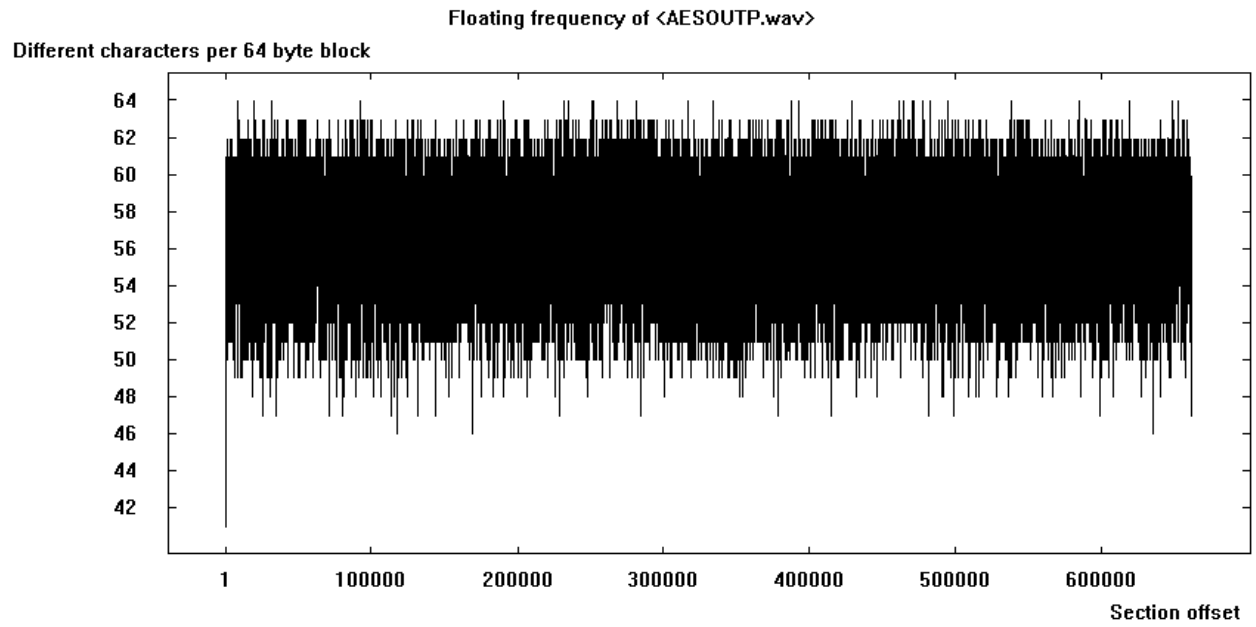
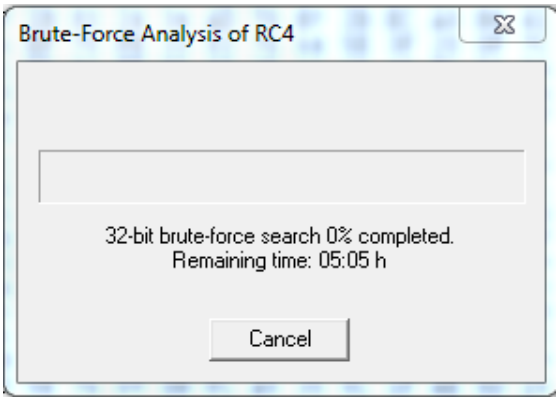


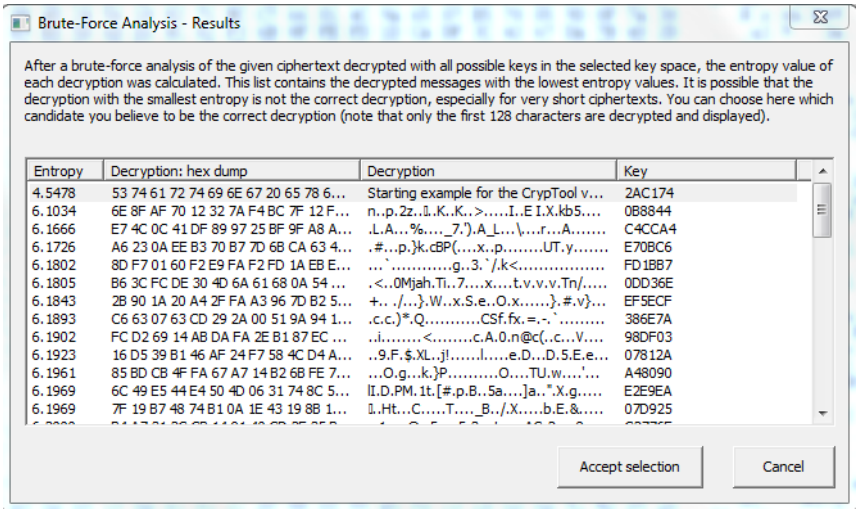
Fig. 6-14 Floating Frequency for Cipher

It is clear from the fig. (6-14) that the floating frequency for the encrypted file is perfect. The most of frequency is between 50 – 65 different characters per 64-byte block compared with fig. (6-13), which ranged between 30-55. This means that there is significant randomness in the newly proposed algorithm, keeping the diffusion in the cipher and leading to more complexity in the relationship between the cipher and the plaintext.

All the figures above and the analysis demonstrate that an important security level has been achieved through the newly proposed encryption algorithm. In addition to the previous analysis, some tests have been carried out, such as brute force attack and poker test, to test the randomness of the output audio file and the strength of the encryption key. These tests have been carried out on the encrypted audio file (cipher) and both of these tests were passed, as shown in Figs (6-15a, 16).



(a) Time



(b)Key Reveal

Fig. 6-15 brute force Attack

The table below shows tests results and illustrates each algorithm with their keys size and the number of possible keys (key space) in addition to the time required to find (recover) the encryption key.

Table 6-9 Key Reveal Time

Cipher	Key size (bit)	Key space	Recovery Time
RC4	32	2^{32}	5 h
AES	128	2^{128}	$2.2*10^{25}$ years
Proposed LEA	128+256	2^{384}	$9.4*10^{64}$ years

The Poker test below also shows the significant level of randomness which has passed the test as shown in the figure

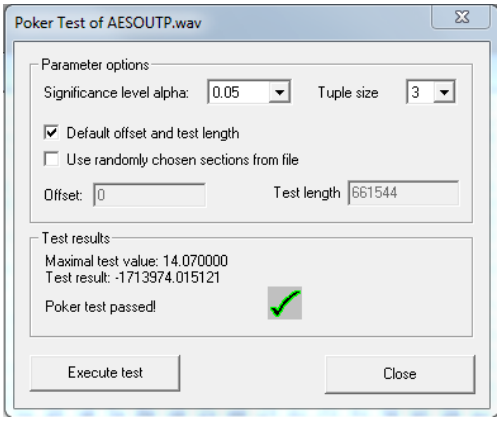


Fig. 6-16 Poker Test Results

6.4 Validation and Evaluation

The validation approach adopts many ways and methods. First, comparison with standard AES algorithm has been carried out. For more validation, it used two different processor specifications, to see how much power saving percentage for each processor.

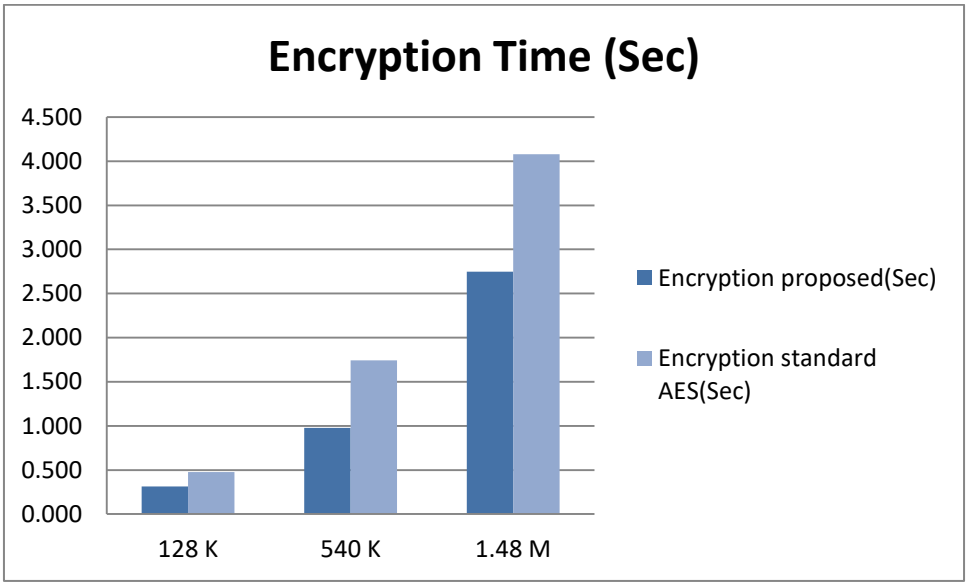
From Tables (6-10, 11) it can be seen that there is a clear reduction in execution time and power consumption for the new algorithm. The amount of reduction reaches nearly 35% for the new algorithm compared with the standard AES.

Table 6-10 Encryption Time compare for LEA and AES

File Size	Encryption Time Proposed (Sec)	Encryption Time Standard AES (Sec)	Energy Saving
128 K	0.314	0.4768	35%
540 K	0.977	1.742	
1.48 M	2.747	4.0776	

Table 6-11 Encryption Energy compare for LEA and AES

File Size	Encryption Energy New algorithm (µ J)	Encryption Energy Standard AES (µ J)	Energy Saving
128 K	0.023	0.035	35%
540 K	0.071	0.109	
1.48 M	0.201	0.298	

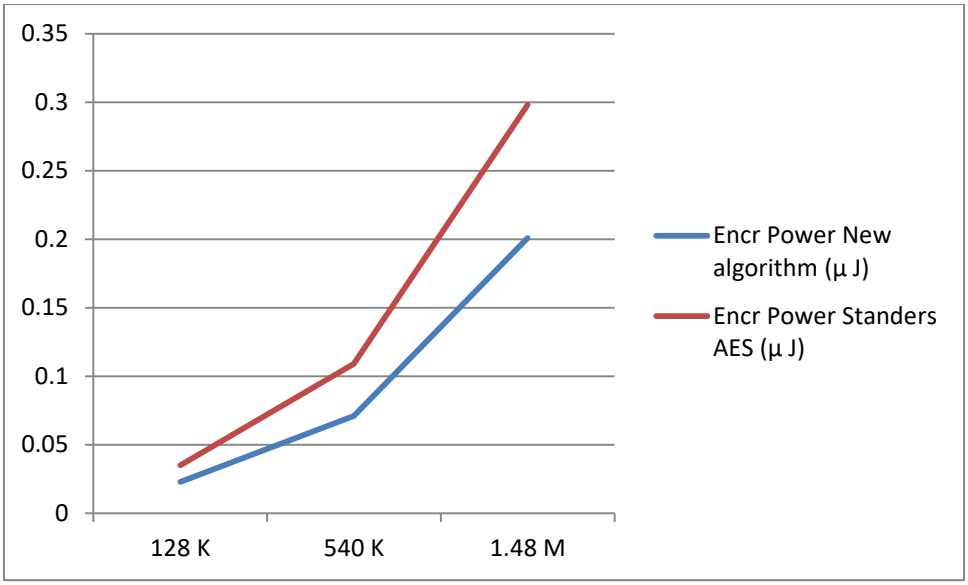


(a)Time

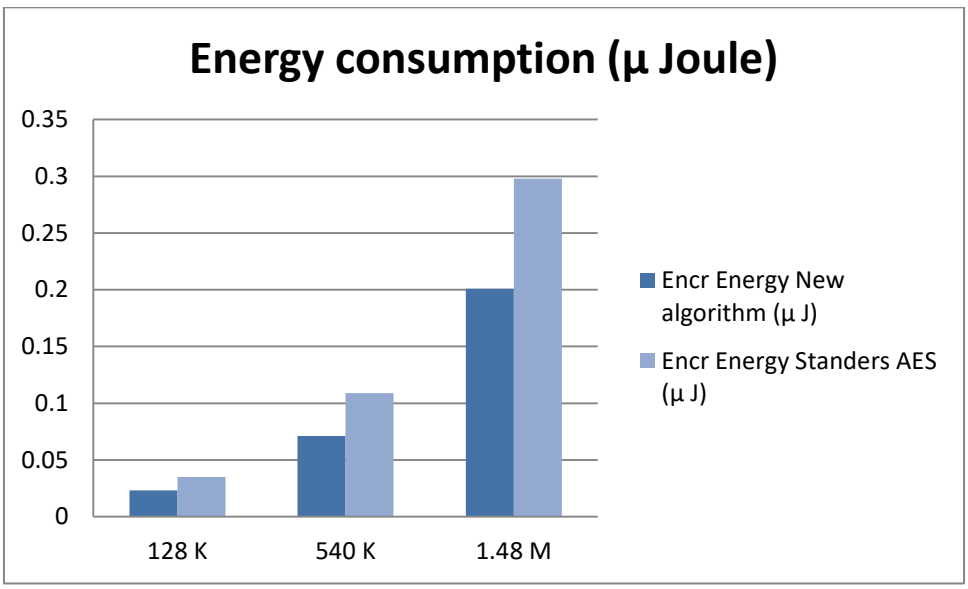
Fig. 6-17 graphs

Let assume, send several long voice messages with *X* sizes using the AES algorithm, and it consumes *E* energy from the battery, so the battery can work for 10 hours.

While in the proposed algorithm it will consume 0.65* E, means 35% extra *E* can be added. If E=10h then new E=13.5 h. means the battery can work extra time roughly 3.5 hours.



(b)Energy



(c)Energy

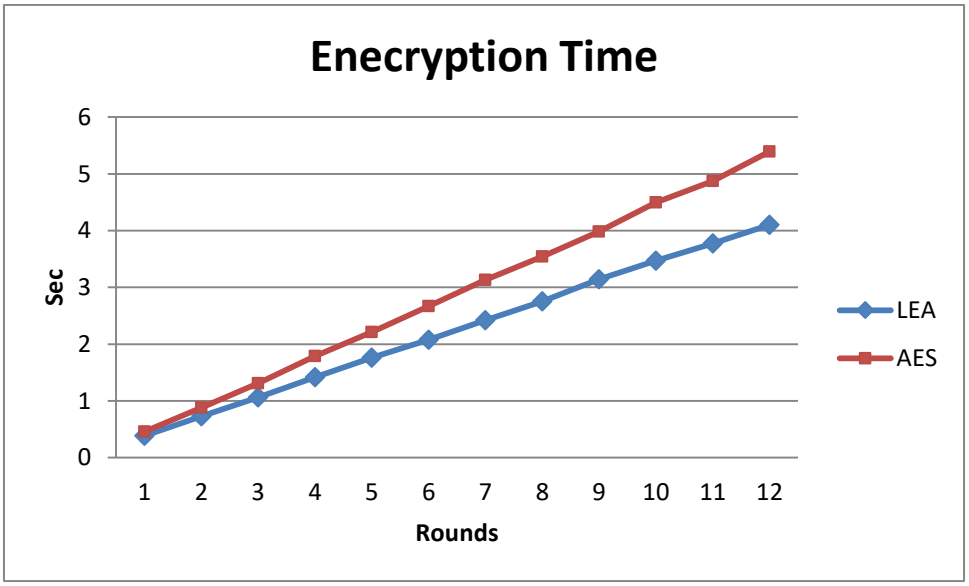
Another comparison between the new LEA algorithm and standard AES has also been done with many numbers of encryption rounds to show the differences between them and their effect on the encryption cost. The new design achieved a better performance in all rounds. The following tables illustrate the results for each round.

Table 6-12 Encryption Time compare for LEA and AES with many Rounds

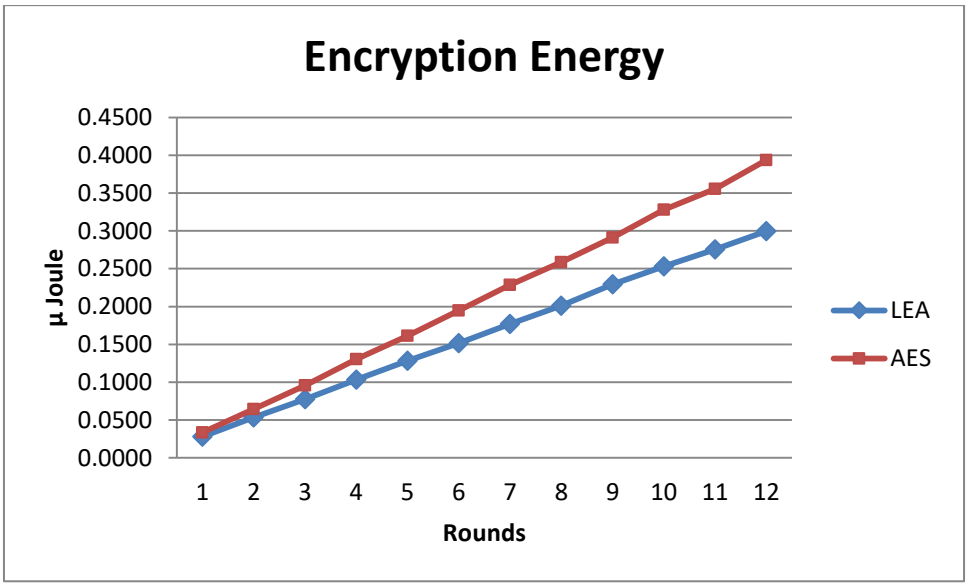
Rounds Iteration	Encryption Time LEA (Sec)	Encryption Time Standard AES (Sec)
1st Rn	0.387	0.459
2nd Rn	0.73	0.884
3rd Rn	1.061	1.313
4th	1.416	1.789
5th	1.759	2.21
6th	2.078	2.669
7th	2.424	3.131
8th	2.755	3.543
9th	3.144	3.986
10th	3.47	4.495
11th	3.773	4.872
12th	4.104	5.392

Table 6-13 Encryption Energy compare for LEA and AES with many Rounds

Rounds Iteration	Encryption Energy LEA (µ J)	Encryption Energy Standard AES (µ J)
1st Rn	0.0283	0.0335
2nd Rn	0.0533	0.0645
3rd Rn	0.0775	0.0958
4th	0.1034	0.1306
5th	0.1284	0.1613
6th	0.1517	0.1948
7th	0.1770	0.2286
8th	0.2011	0.2586
9th	0.2295	0.2910
10th	0.2533	0.3281
11th	0.2754	0.3557
12th	0.2996	0.3936



(a)



(b)

Fig. 6-18 Statistics for Many Rounds

The results show that significant improvements in the proposed algorithm were achieved. The significance of the results is measured using a T-test function. The critical value for this test is (0.05). The output of the test shows that the *P-value < 0.05.*

In the quality test: *P-value = 0.000387 < 0.05*

For security test: *P-value = 0.000206 < 0.05*

So this results are consider as significant and the algorithm has achieved a good tradeoff in security and quality.

Table (6-14) shows the comparison of features for both algorithms. It can be seen that there are many important differences between them. The performance is batter and the security is maintained and increased in some parts. Also, it is clear that the file size is still the same, which will therefore not affect the memory size and the bandwidth of network paths.

Table 6-14 Features comparison between the Standard AES and the proposed LEA

Property	**AES**	**New Algorithm**
S_box	Single	Multi
Num. of keys	1	2
Keyspace	2^{128}	$2^{128}*2^{256}*8!$
Binary Histogram	Random	Random
Poker Test	Pass	Pass
Security	Secure	Secure
Block length	16 Bytes	same
No. Rounds	10	9
Encryption time	**T**	**0.65 T**
Power consumption	**P**	**0.65 P**
Energy Saving	---	**35%**
Output = input size	Yes	Yes

From all the above evaluation, there is a good performance has been achieved by new proposed algorithm LEA making it cost-effective and energy saver. Also, a significant level of security has been achieved by the new algorithm. The algorithm LEA has achieved a good tradeoff in security and quality.

6.5 Summary

A lightweight and low energy encryption algorithm for audio files was developed and tested in this chapter. The main objectives have been achieved by reducing the execution time and energy consumption of the encryption process compared with the standard algorithm (AES) and keeping its security level in a good complexity.

A Low computation Mixcolumn function and nine rounds iteration for the new algorithm LEA have been proposed in this chapter. The testing and experiments were conducted and a range of implementation scenarios are setup with different audio files. Also, the algorithm has been tested with many iteration rounds to test their performance and effect. The test results show significant improvements in new design metrics. The comparison between the new algorithm and the standard one shows a significant amount of time and energy consumption reduction being achieved (approximately 35%)

Data security was analyzed using specific testing tools, to measure the new algorithm strength. Many security parameters have been tested such as binary histogram, autocorrelation, and others to test the randomness and the complexity of the cipher.
The validation and evaluation showed that a significant level of security has been achieved by the new algorithm. In addition, a good performance making it cost-effective and energy saver. The algorithm LEA has achieved a good tradeoff solution in security and quality. The new design is more suitable for the wireless environment and helps to address the limitation of the wireless devices.

The next chapter will use the outcome and the proposed functions of this chapter, to develop a novel encryption algorithm with new features.

7 Chapter seven: A Novel Triple Key Encryption Algorithm (TKE)

7.1 Introduction

This chapter will make further development for the algorithm that has already been proposed in chapter 5 and 6, to increase the security level by adding 3rd key function. The SubByte function proposed in chapter 5 and mixcol proposed in chapter 6, in addition, the 3rd key function will be all used to propose the novel encryption algorithm for high security and lightweight consumption. Again, there are two factors that can help to increase the complexity of any cryptography algorithm (Alsalami, et al., 2016), which are the Confusion and Diffusion, the strong encryption needs much more confusion and diffusion.

Research objective #5 was built upon claims by (Abhiram, et al., 2015) that weakness of AES is that it works with a single key. Also, the recommendation of NIST about the key importance and key management (Scarfone, et al., Novmber 2007), gave further attention on key security because the confidentiality of the key determines the security of the algorithm (Alamsyah, et al., 2017). For instance, man in the middle attack can fraudulently capture the cryptography key and use it to reveal the encrypted data. Therefore, a third key has been added to the proposed algorithm to increase the security level for it, this key is XOR with the output ciphertext in the last round only.

An encryption algorithm is going to be investigated and developed in this chapter to increase the complexity of encryption process, reduce the execution time and use the additional key, making it difficult to breach and more resistance to many attacks such as differential attacks, and in the same time decrease the execution time and energy consumption. The base of this work is the AES algorithm. As explained in previous chapters, A SubByte function using multi S-box transformation technique has been developed to increase the confusion and complexity of encryption algorithm, because each byte in state block will substitute with another Byte from S-Box table (Hazzaa, et al., 2018). Also, developing the Mix-column operation and proposing 9 rounds iteration would make the algorithm lightweight and decrease the power consumption as explained in chapter six. To increase security, the 3rd key would be proposed in this chapter. All of these proposed functions would produce a Novel scheme to encrypt the audio files which are useful in the wireless environment when the QoS is crucial.

There is a lack of researches that dealing with such a technique. However, there are roughly similar works that could address some issues in encryption of voice in wireless networks connected to the Internet. (Alamsyah, et al., 2017) built S-box using a basic polynomial equation and the addition of a constant 8-bit vector different from the standard AES. (Ali, et al., June 2014), (Mohammed & Rohiem, 2009) used Dual keys technique which helped to increase the complexity of the algorithm. However, there is a lack of convincing security argument which proves its strength. This chapter will address these gaps by testing and analysing the new encryption algorithm. (RAMESH & UMARANI, 2012) also, states that the key size has had some effect on the performance. They compared the performance of changing different key sizes for AES. They consider three possible key sizes i.e., 128 bit, 192 bits and 256-bit keys in AES. The results show that higher key size leads to a clear change in the battery and time consumption. It can be seen that going from 128 bits key to 192 bits causes an increase in power and time consumption about 9% and to 256-bit key causes an increase of 17%. So, choosing the key size is considered in this research.

Aim

- To propose a novel encryption algorithm with triple key and high level of complexity and at the same time reduce the execution time and power consumption
- Add 3rd key function
- Using the NIST test suite to analyze the security
- Evaluate and analyze the security and QoS parameters to prove their strength.

Methodology

In this chapter, a quantitative research method has been adopted which involves running two security encryption experiments for audio files with deferent sizes. A proposed algorithm has been explained and tested. As mentioned in the previous section, three functions for encryption algorithm will be developed. The delay time and Energy consumed parameters have also been measured. A security analysis has been conducted to prove the algorithm strength. The testing has been carried out using Visual Studio 2015 with C++ programing language. The proposed TKE algorithm has been tested on an audio file with different sizes. Finally, the evaluation and comparison with the standard algorithm have been carried out.

A security analysis has been conducted to test the security level of the new algorithm. Many security parameters have been measured, such as the randomness of the encrypted data like Entropy, Histogram, Autocorrelation, and Floating frequency test. The poker test and frequency test are carried out as well. The CrypTools and NIST statistic suite for

cryptography and cryptanalysis have been used to carrying out these tests. All the tests have conducted on audio file format (.wav) with different sizes.

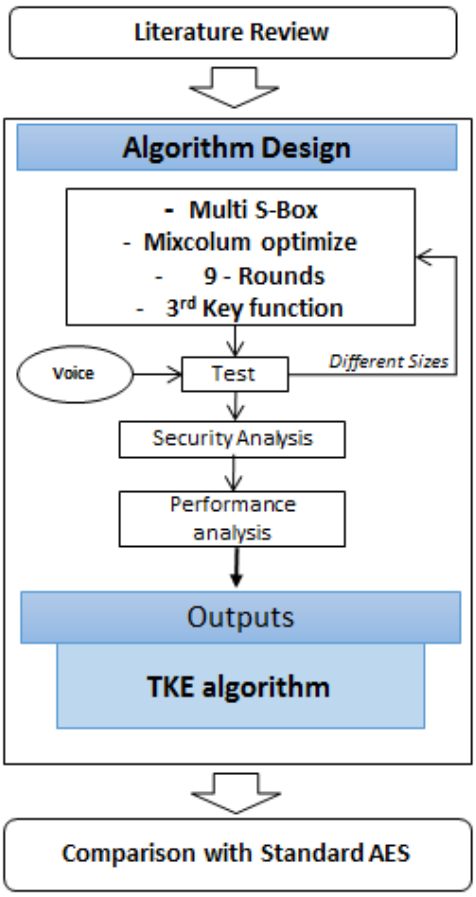


Fig. 7-1 Methodology Design

7.1.1 Key in AES

In cryptography, the key is a variable value that is applied using an algorithm to a block of plain text to create encrypted text and use the same key to decrypt the encrypted text. The length of the key is a factor in considering how hard it will be to decrypt the text in a given message (Stallings, 2017).

Unlike symmetric key algorithms that rely on one key to both encrypt and decrypt, each key performs a unique function (Goodrich & Tamassia, 2011). The public key is used to encrypt and the private key is used to decrypt. It is computationally infeasible to compute the private key based on the public key.

A cryptographic algorithm, or cipher, is a mathematical function used in the encryption and decryption process (Stallings, 2012). A cryptographic algorithm works in combination with a key to encrypt the plaintext. The same plaintext encrypts to different cipher-text with different keys.

An encryption key is usually a random string of bits produced specially to scramble and unscramble data. Encryption keys are created with algorithms designed to ensure that each key is unique and unpredictable. 128-bit AES keys are symmetric keys.

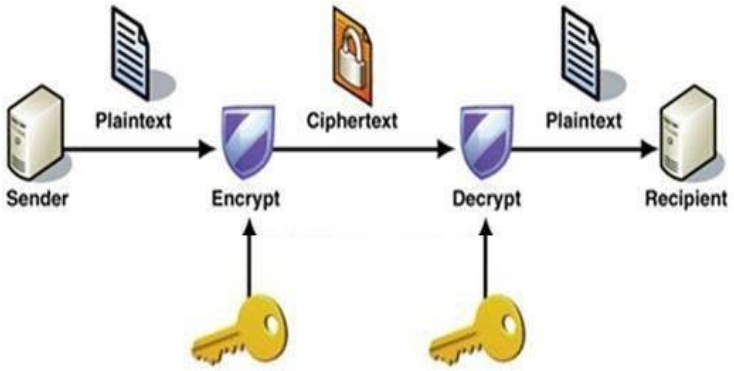


Fig. 7-2 Cryptography Steps

7.2 Proposed Triple Key Encryption Algorithm (TKE)

The TKE is composed of three achievements:

7.2.1 Proposed SubByte transformation function

This function has already been explained in chapter 5 and will be used here

7.2.2 Proposed the lightweight functions

This function has already been explained in chapter 5 and will be used here

7.2.3 The proposed 3rd key function

The AES algorithm uses the key to encrypt and decrypt the data. The importance of the key is very sensitive; the key should be secret between the sender and the receiver. If the attacker knows the key, he will decrypt the cipher quickly and easily, even if the algorithm is very complex. So, increasing the complexity of the algorithm is not enough, the number of keys also need to be increased in order to achieve a high level of security.

However, increasing a huge number of the keys may not always work in wireless networks, because it will add more load to the network when using key exchange protocol for exchanging the keys between the nodes. So, this should be taken into consideration when planning to increase the number of keys.

In this section, a third key has been added to the proposed algorithm in order to increase the security level for it, this key is XOR with the output ciphertext in the last round only. The third key length is 16-byte, thus the key space for it is 128! , and accordingly, the complexity

in the proposed algorithm is increased by as much as the total key space added to the above algorithm. The following demonstrates the total keyspace, or added complexity.

$$New\ Key\ complexity\ =\ 2^{128}$$

So, the Number of possible keys:

$$2^{128} = 3.4 * 10^{38}$$

$$T\ =\ 5.3 * 10^{21}\ years$$

Where T = time to find the key

The probability of finding three keys is very rare. The third key has been added in the last round of new algorithm not in all rounds, to keep the execution time and power consumption at an acceptable level and not increasing it too much. Algorithm 5 represents the modification in the last round and adding the new key.

Algorithm 1 Final algorithm TKE

I/P: plaintext Block {State [Row][Col]Row, Col=1,2,3,4}. With 1-16 S-boxes, generated in algorithm 1.
O/P: cipher text C {[Row][Colum]Row Colum=1,..,4}

Encrypt using nine rounds
for (i = 0; i<8; i++)
SubBytes(); // subbytes transformation proposed in Chapter (5)
ShiftRows();
MixColumns(); // mixcol proposed in Chapter (6)
AddRoundKey();
Last round;
End.

Algorithm 2 Proposed last round

Last round :
SubBytes(); // subbytes transformation proposed in Ch.5
ShiftRows();
AddRoundKey();
AddRoundKey3();

Fig (7-3) illustrates the new proposed Triple key process. There are three keys used in the new algorithm for encryption/decryption operation. The first one is the main key, the second

is the S-box generation key and the third is the last round key. This is will definitely address the brute force attach and man in the middle attack.

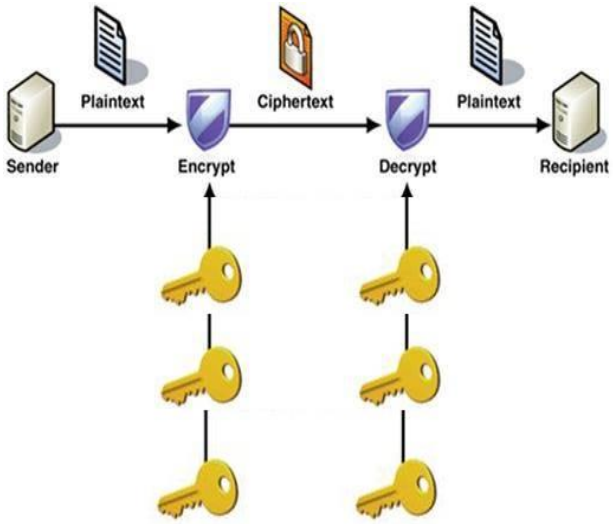


Fig. 7-3 Proposed Cryptography Steps

7.3 The Overall Novel Design Framework

The new design has added new features to the functions of the algorithm, as stated in chapter 5,6 and 7, which make a reliable tradeoff between the security and QoS parameters. The power and time consumption have been reduced as explained in chapter 5 and 6. and the security has been increased in 5 and 7. The new overall complexity of the new algorithm will be as follow:

$$Triple\ Key: \quad 2^{128} * 2^{256} * 8! * 2^{128} \qquad (5)$$

The attacker needs to know three keys at the same time, and it is very difficult to guess all of them together; so, even if one key is hacked, it will be more complicated to find another. Also, theoretically, if the attacker uses the Brute-force attack to find all the keys, then a huge time will be required, greater than $5.3*10^{21}$ years (Hazzaa, et al., 2018). Therefore, in this case, the security level for this algorithm will be maintained or increased.

As mentioned before, the new algorithm is more suitable for a wireless environment because of the new features. The following section provides more analysis linked back to these facts. Fig (7-4) demonstrates the proposed enhancement in the new algorithm. The modified functions are new SubByte and new MixColumn. In addition, the function Addroundkey3 was added in the last round, to increase the security level.

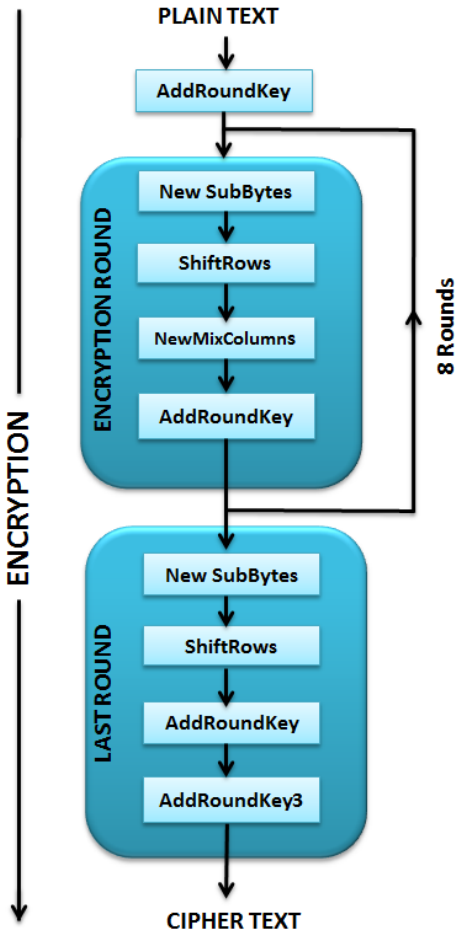


Fig. 7-4 New TKE Algorithm Design (Author)

7.4 Experiment

This test aims to execute the encryption process on the audio file using the proposed algorithm (Triple Key Encryption algorithm TKE), to determine the execution time and power consumption. Two scenarios have been conducted in Lab's wireless laptop and computer with window 7 using a different kind of processer. In the first scenario (quad-core i7, Ram 8 GB) used. The other scenario (Due Core, Ram 4 GB) and the proposed AES algorithm have been tested on an audio file with different sizes, and each test has been run for 5 times for accuracy.

The program code will call the file and open and read it. Then execute the encryption process by reading and encrypting it block by block, each block has 16 byte.

The proposed algorithm uses Triple keys; the first key is the main key which is used in encryption process in each round as in AES algorithm. The second key consists of two parts.

The first part a set of multi-values up to 16 elements. Each value in the key set has another value related to it, leading to building different S-boxes with its related inverse S-Box. **Rndom_Key[8]={0x67, 0x85, 0x25,0xb5,0xA4, 0xf1,0x19,0x4c} and Cons_c[8]={ 0x82, 0x45, 0xc4, 0xa5,0x7b, 0x63,0xd5,0xc1}** represent the first key, based on hexadecimal, each value in the key with its related cons_c value, can create unique S-Box. The second part randomly distributes the S-boxes which created s box key as explained in chapter five.

Table 7-1 S-Box generated by K: 0x67 and C: 0x82

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	82	4f	F0	DE	2f	B6	AC	06	C0	FE	98	17	01	63	54	76
1	A3	36	28	C9	1B	03	48	22	43	E8	E6	4E	7D	F1	6C	9A
2	12	8A	D8	BF	D7	65	B3	B4	DA	77	D6	A4	E7	C6	46	8C
3	62	0B	23	11	24	44	E4	6A	E9	1D	3B	47	F5	39	8E	FD
4	CA	B0	86	29	AF	2A	88	EB	BC	7F	E5	08	1A	C1	0D	21
5	3A	74	78	E2	A8	0C	05	0E	30	25	A0	72	E0	F7	85	3F
6	F2	EF	D2	9D	52	71	4B	A7	45	90	75	C4	B1	EE	F6	DF
7	37	60	D9	9E	5E	FB	F4	BE	AD	94	CB	2A	10	87	A9	FF
8	32	56	9B	64	14	C2	C3	81	80	7A	42	68	13	19	A2	EA
9	09	BD	7C	DC	A5	91	53	0F	CE	69	B7	0A	D1	92	C7	4D
A	4A	CD	F9	41	6B	6F	B2	9F	97	79	C5	04	B5	CF	50	D3
B	DB	AE	51	A1	93	6E	FA	59	27	A6	38	73	95	58	C8	4C
C	BA	55	34	8B	3E	FC	99	8D	7E	5A	7B	B5	66	2B	84	02
D	61	E3	1F	IE	ED	F3	35	5B	8F	5C	20	31	2C	1C	B8	70
E	CC	16	67	96	BB	40	18	49	EC	33	AA	F8	B9	2D	9C	57
F	15	6D	89	D0	26	5D	D4	3D	5F	E1	00	DD	83	AB	3C	07

Table 7-2 Inverse S-Box generated by K: 0x67 and C: 0x82

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	FA	0C	CF	15	AB	56	07	FF	4B	90	9B	31	55	4E	57	97
1	7C	33	20	8C	84	F0	E1	0B	E6	8D	4C	14	DD	39	D3	D2
2	DA	4F	17	32	34	59	F4	D8	12	43	45	CD	BC	ED	7B	04
3	58	DB	80	E9	C2	D6	11	70	BA	3D	50	3A	FE	F7	C4	5F
4	E5	A3	8A	18	35	68	2E	3B	16	E7	A0	66	BF	9F	1B	01
5	AE	B2	64	96	E	C1	81	EF	BD	B7	C9	B7	D9	F5	74	F8
6	71	D0	30	D	83	25	CC	E2	8B	99	37	A4	1E	F1	B5	A5
7	DF	65	5B	BB	51	6A	F	29	52	A9	89	CA	92	1C	C8	49
8	88	87	00	FC	CE	5E	42	7D	46	F2	21	C3	2F	C7	3E	D8
9	69	95	9D	D4	79	BC	E3	A8	A	C6	1F	82	EE	63	73	A7
A	5A	B3	8E	10	2B	94	B9	67	54	7E	EA	FD	06	78	B1	44
B	41	6C	A6	26	27	CB	05	9A	DE	EC	C0	E4	48	91	77	23
C	08	4D	85	86	6B	AA	2D	9E	BE	13	40	7A	E0	A1	98	AD
D	F3	9C	62	AF	F6	AC	2A	24	22	72	28	B0	93	FB	03	6F
E	5C	F9	53	D1	36	4A	1A	2C	19	38	8F	47	E8	D4	6D	61
F	02	1D	60	D5	76	3C	6E	5D	EB	A2	B6	75	C5	3F	09	7F

Fig (7-5) shows the hexadecimal representation of the input file before the encryption (Plain).

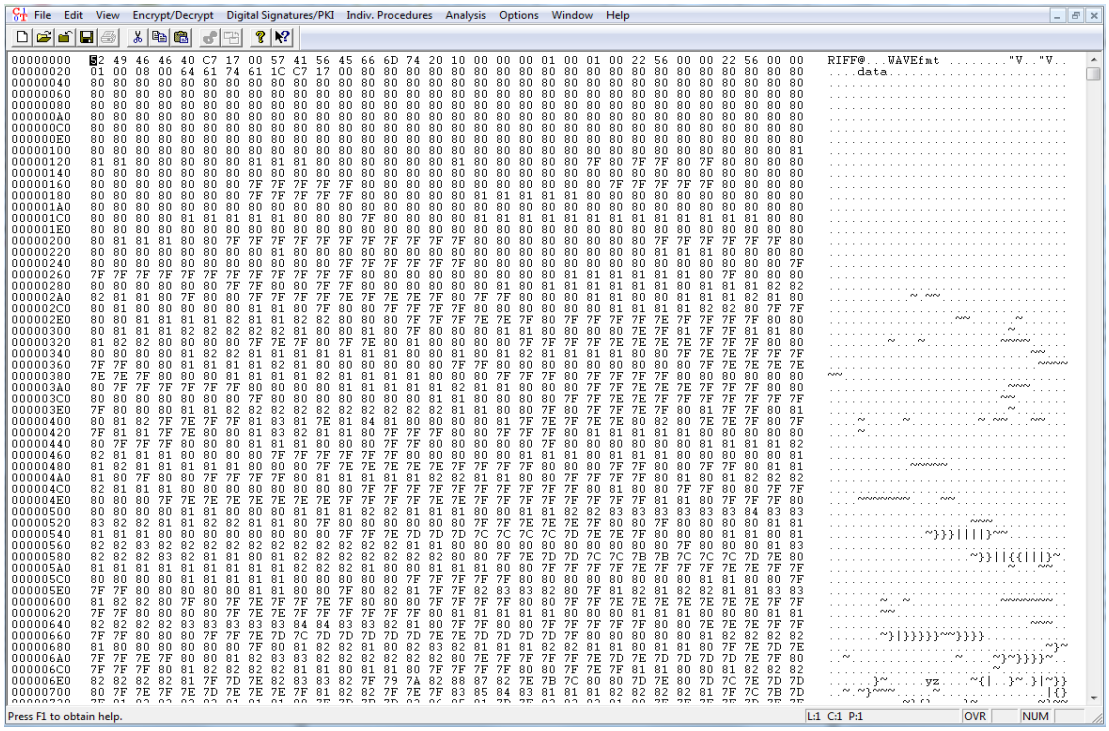


Fig. 7-5 Plain File

The third key will be added in the last round and the length is 16 byte. The following code illustrates the encryption process with the new functions:

TKE Encryption

```
void ENCRYPT(void)
{

	unsigned char r, c;
	unsigned char i;
	AddRoundKey();
	for (i = 0; i<8; i++)
		MainEnc();
	SubBytes();
	ShiftRows();
	AddRoundKey();
for (r = 0; r<4; r++)
	{
		for (c = 0; c<4; c++)
		{
			State[r][c] ^= key3[r][c];
		}
	}
	k = 0;
	return;
}
```

After executing the encryption program (source.cp) by the software, the command line appears and asks to enter the main encryption key. The key used in this excrement is:

Main Key = fedcba987654321abcdef123456789ff

The length of the key is 32 char, means 16 bytes, 128 bit.

The 3rd key used in the last round is:

```
unsigned char key3[4][4] = { 0x02,0x13,0xa1,0xb2,0xc3,0xd4,0xe6,0xf7,
0x89,0x75,0x34,0x56,0x78,0x01,0x02,0x03 };  //3rd key to increase the security
```

After entering the key and hit enter, the execution start to encrypt the file. The following code executes the encryption process on the input audio file:

```
for (i = 0; i<i_count; i += 16)
        {
                fread(State, sizeof(char), 16, Rfile);
                ENCRYPT();
                fwrite(State, sizeof(char), 16, Wfile);
        }
```

Fig (7-6) shows the screenshot of the execution of this experiment and show the main key used in the encryption process.

The other scenario is testing with different rounds iteration from 1---12 round to test their effect on the performance metrics.

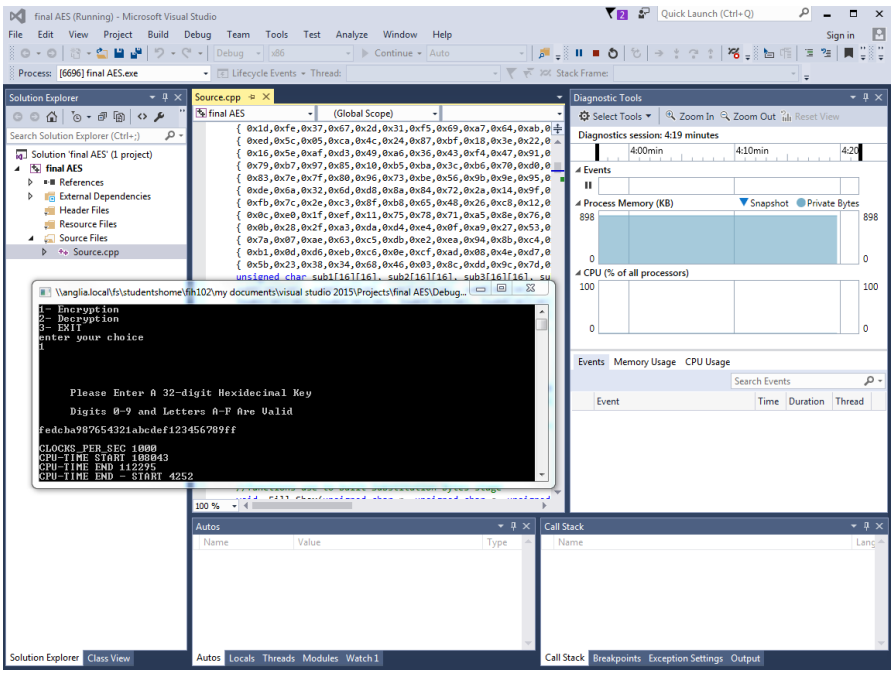


Fig. 7-6 Snapshot of the Test

7.4.1 Experimental Results

This section explains the results of the above experiment and shows the figures and numerical data for the outcome of the proposed scheme. Fig (7-7) show the output file data after the encryption. The interpretation of such Hex figure described in chapter two.

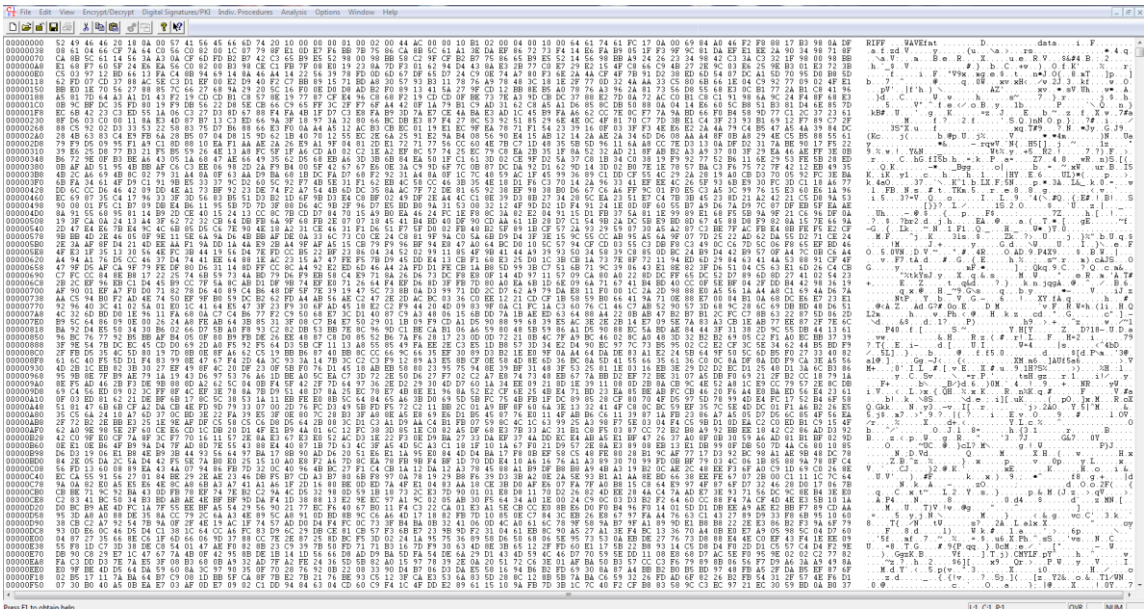


Fig. 7-7 The Cipher file

The following tables show the time is taken and the power consumed by the standard and the new algorithm for the encryption and decryption process.

The table (7-3) shows the average execution time for 5 repeating execution of the program (more results tables can be found in the appendix).

Table 7-3 average of execution time

128K	Encry	Decry	540 K	Encry	Decry	1.48 M	Encry	Decry
	0.328	0.335		0.98	1.015		2.762	2.864
	0.318	0.329		0.986	1.014		2.755	2.857
	0.329	0.329		0.981	1.014		2.759	2.879
	0.326	0.326		0.978	1.011		2.76	2.85
	0.325	0.327		0.979	1.012		2.76	2.86
Average	0.3252	0.3292	**Average**	0.9808	1.0132	**Average**	2.7592	2.862

Table (7-4) illustrates the amount of the execution time and Energy which has been taken by both the proposed and the standard AES algorithm to encrypt and decrypt the audio files with different size using (quad-core i7, Ram 8 GB) processer. In the proposed algorithm, the highest level it reached nearly 2.759 sec with 1.4 MB while the lowest is 0.33 sec with 128 KB file size.

Table 7-4 Cryptography Process using (quad-core i7)

(a)execution time and Energy by TKE

	Execution Time TKE		Energy Consumption	
File Size (B)	**Encryption Time (Sec)**	**Decryption Time (Sec)**	**Encryption Energy (µ J)**	**Decryption Energy (µ J)**
128 K	0.33	0.329	0.024	0.024
540 K	0.981	1.013	0.072	0.074
1 M	1.857	1.9	0.135	0.138
1.48 M	2.759	2.862	0.201	0.209

(b) Execution time and Energy by AES

	Execution Time AES		Energy Consumption	
File Size (B)	**Encryption Time (Sec)**	**Decryption Time(Sec)**	**Encryption Energy (µ J)**	**Decryption Energy (µ J)**
128 K	0.477	0.462	0.035	0.034
540 K	1.493	1.387	0.109	0.101
1 M	2.787	2.668	0.2	0.195
1.48 M	4.078	3.885	0.298	0.284

Table (7-5) illustrates the amount of the execution time and Energy which have been taken by both the proposed and the standard AES algorithm to encrypt and decrypt the audio files with different size using (Due Core, Ram 4 GB) processer. In the proposed algorithm, the highest level it reached approximately 4.13 sec with 1.4 MB while the lowest is 0.51 sec with 128 KB file size.

Table 7-5 Cryptography Process using (Due core)

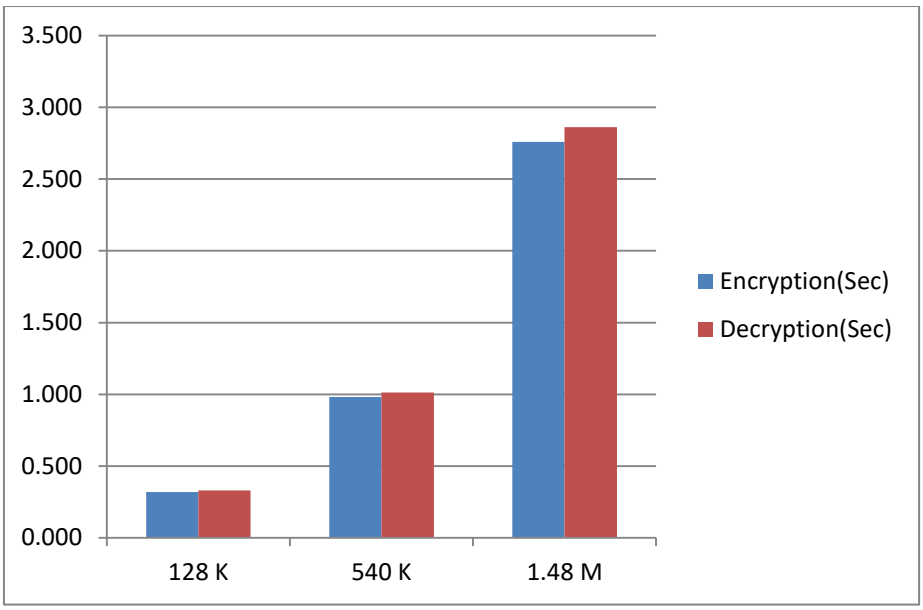
(a) Execution time and Energy by TKE

File Size (B)	Encryption Time (Sec)	Decryption Time (Sec)	Encryption Power (µ J)	Decryption Power (µ J)
128 K	0.51	0.58	0.037	0.04
540 K	1.425	1.798	0.1	0.131
1 M	3.21	3.29	0.234	0.24
1.48 M	4.13	4.502	0.3	0.33

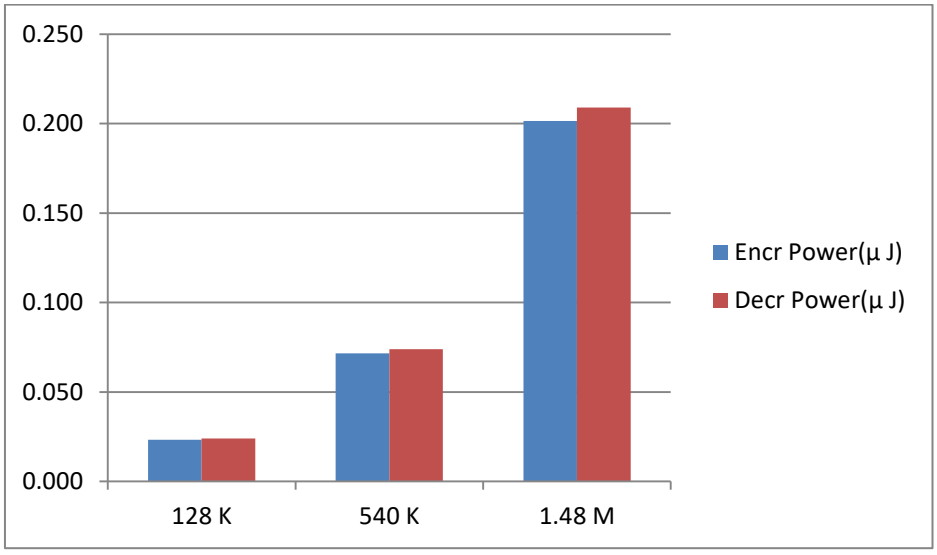
(b) Execution time and Energy by AES

File Size (B)	Encryption Time (Sec)	Decryption Time (Sec)	Encryption Power (µ J)	Decryption Power (µ J)
128 K	0.67	0.74	0.049	0.054
540 K	2	2.2	0.14	0.16
1M	4.41	4.597	0.321	0.335
1.48 M	6.2	7.5	0.45	0.55

The statistics figure below show the characteristic of encryption and decryption process in TKE algorithm, it is clear that the same amount of time for both processes.



(a)Execution Time



(b)Energy Consumed

Fig. 7-8 Graphs

Also, a different key is used: **Main Key = abcdef123456789abcdef123456789ac,** the length of the key is 32 char, means 16 byte, 128 bit. There are no clear differences between the two scenarios

Table 7-6 Different Key Encryption

	Execution Time		Energy Consumption	
File Size (B)	**Encryption Time (Sec)**	**Decryption Time (Sec)**	**Encryption Energy (µ J)**	**Decryption Energy (µ J)**
128 K	0.33	0.33	0.024	0.024
540 K	0.98	1.01	0.072	0.074
1.48 M	2.76	2.86	0.201	0.209

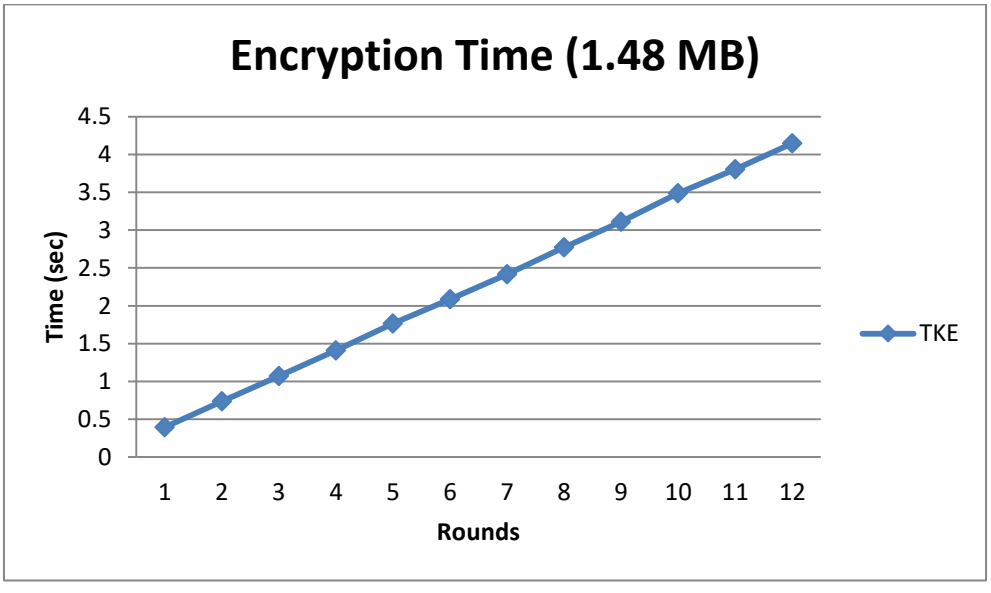
The experiment has also tested the new proposed algorithm with many numbers of encryption rounds to show the differences between them and their effect on the encryption cost. The following tables illustrate the results for each round.

Table 7-7 Execution Time with many rounds

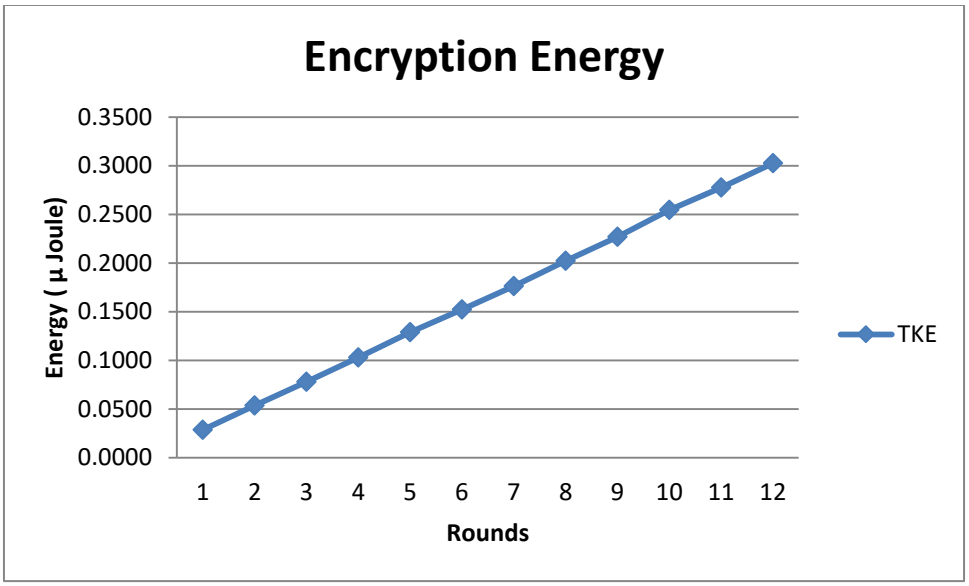
Rounds Iteration	**Encryption Time (Sec)**	**Decryption Energy (µ J)**
1st Rn	0.393	0.398
2nd Rn	0.736	0.74
3rd Rn	1.07	1.1
4th	1.412	1.417
5th	1.765	1.77
6th	2.087	2.1
7th	2.417	2.42
8th	2.774	2.77
9th	3.109	3.1
10th	3.487	3.48
11th	3.804	3.81
12th	4.148	4.15

Table 7-8 Energy consumption with many rounds

Rounds Iteration	Encryption Energy (µ J)	Decryption Energy (µ J)
1st Rn	0.0287	0.0291
2nd Rn	0.0537	0.0540
3rd Rn	0.0781	0.0803
4th	0.1031	0.1034
5th	0.1288	0.1292
6th	0.1524	0.1533
7th	0.1764	0.1767
8th	0.2025	0.2022
9th	0.2270	0.2263
10th	0.2546	0.2803
11th	0.2777	0.2781
12th	0.3028	0.3030



(a)



(b)

Fig. 7-9 Statistics of Many Rounds for 1.48 M

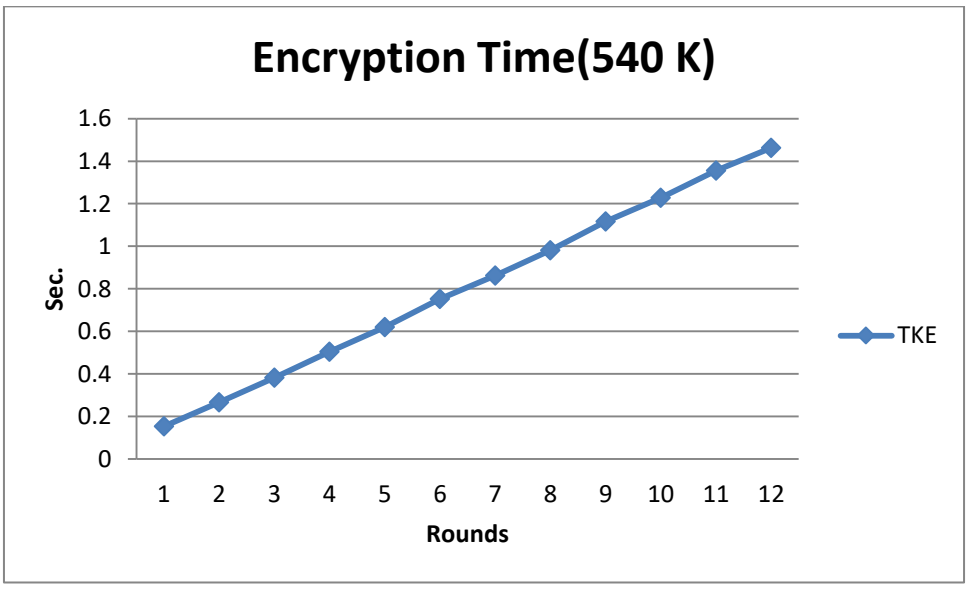
Table 7-9 Encryption for (teaching.wav) file with many Rounds

(a) Execution Time

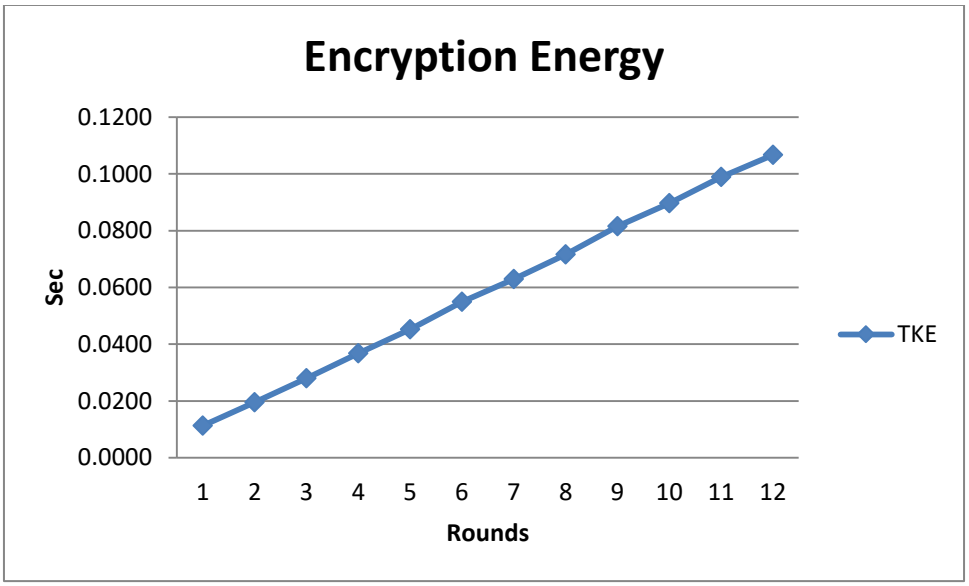
Rounds Iteration	Encryption Time (Sec)	Decryption Time (Sec)
1st Rn	0.154	0.154
2nd Rn	0.267	0.27
3rd Rn	0.383	0.384
4th	0.504	0.507
5th	0.62	0.621
6th	0.752	0.75
7th	0.863	0.864
8th	0.981	0.981
9th	1.117	1.119
10th	1.228	1.227
11th	1.355	1.355
12th	1.462	1.463

(b) Energy Consumption

Rounds Iteration	Encryption Energy (µ J)	Decryption Energy (µ J)
1st Rn	0.0112	0.0112
2nd Rn	0.0195	0.0197
3rd Rn	0.0280	0.0280
4th	0.0368	0.0370
5th	0.0453	0.0453
6th	0.0549	0.0548
7th	0.0630	0.0631
8th	0.0716	0.0716
9th	0.0815	0.0817
10th	0.0896	0.0896
11th	0.0989	0.0989
12th	0.1067	0.1068



(a)



(b)

Fig. 7-10 Statistics of Many Rounds for 540 K

7.5 Validation and Analysis

The validation approach adopts many ways and methods. First, comparison with standard AES algorithm has been carried out. For more validation, we used two different processor specifications, to see how much power saving percentage for each processor. Also, we categorized into two categories, QoS and security. Furthermore, we used different file patterns to check their effect. We can also mention that the recording encrypted\decrypted files are easy to hear by human and it proves the successful implementation of our experiment.

7.5.1 Time and Power

Here we explain the time and power consumed in our experiment and compared it with standard AES ina different processor.

Table (7-10) shows the execution outcome for the audio file with different patterns. The result showed that no clear difference between them. So the pattern didn't affect the QoS in the proposed algorithm.

Table 7-10 Execution outcome for different Patterns

File Size (B)	Voice pattern	Encryption Time (Sec)	Decryption Time (Sec)	Encryption Power (µ J)	Decryption Power (µ J)
540 K	Lady Phone call	1.424	1.796	0.1	0.131
540 K	Man lecture	1.421	1.789	0.1	0.131
540 K	Music sound	1.423	1.791	0.1	0.131

The following tables illustrate the results of implementing our proposed algorithm using two different processors. As explained in ch.3, the power factor = 0.073

Table 7-11 Comparison for TKE and AES (quad-core i7)

(a) Time

File Size (B)	Encryption Time Standard (Sec)	Encryption Time Proposed (Sec)	Improved percentage (%)
128 K	0.477	0.33	30%
540 K	1.493	0.981	31%
1.48 M	4.078	2.759	32%

(b) Energy

File Size (B)	Encryption Energy Standard (µ J)	Encryption Energy Proposed (µ J)	Avr. Improved percentage (%)
128 K	0.035	0.024	30%
540 K	0.109	0.072	
1.48 M	0.298	0.201	

Table (7-12) illustrates the differences between the proposed algorithm and the standard AES implemented on processor (quad-core i7, Ram 8 GB). While table (7-12) compare between the proposed algorithm and the standard AES implemented on processor (Due Core, Ram 4 GB).

In both devices there are roughly the same improvement percentage has achieved which are 30%, this means that the proposed algorithm has consumed by 30% less power and time wherever it would implements. This validation approved the enhancement of the proposed encryption algorithm.

Also, the comparison between the standard and proposed algorithm has shown an average of 30% improvement in the time and the energy.

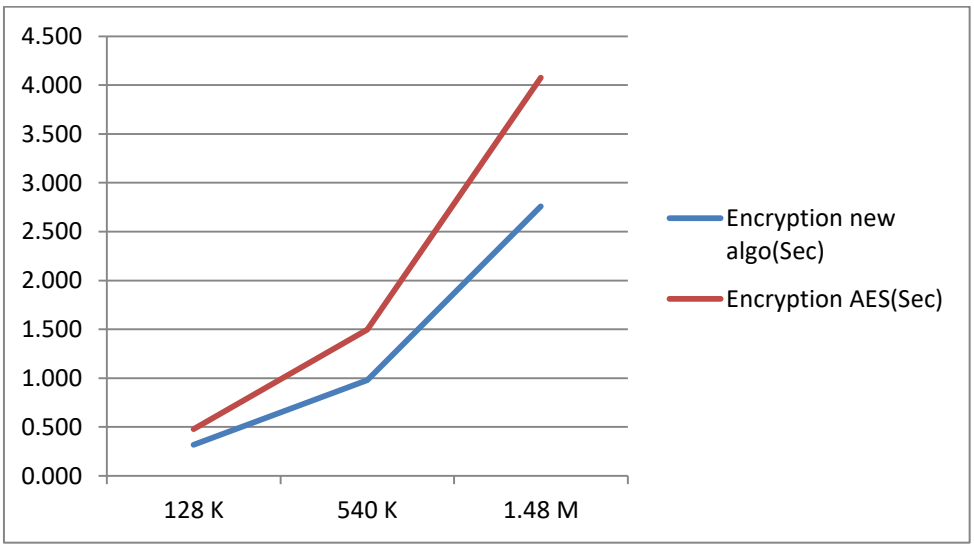
Table 7-12 Comparison for TKE and AES (Due core)

(a) Time

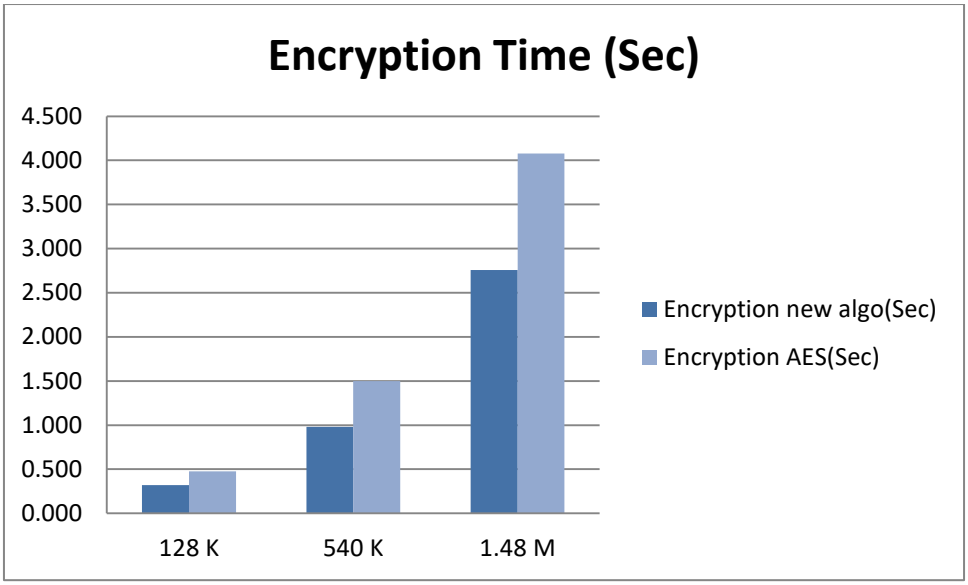
File Size (B)	Encryption Time Standard (Sec)	Encryption Time Proposed (Sec)	Avr. Improved percentage (%)
128 K	0.67	0.51	30%
540 K	2	1.425	
1.48 M	6.2	4.13	

(b) Energy

File Size (B)	Encryption Power Standard (µ J)	Encryption Power Proposed (µ J)	Avr. Improved percentage (%)
128 K	0.049	0.037	30%
540 K	0.14	0.1	
1.48 M	0.45	0.3	

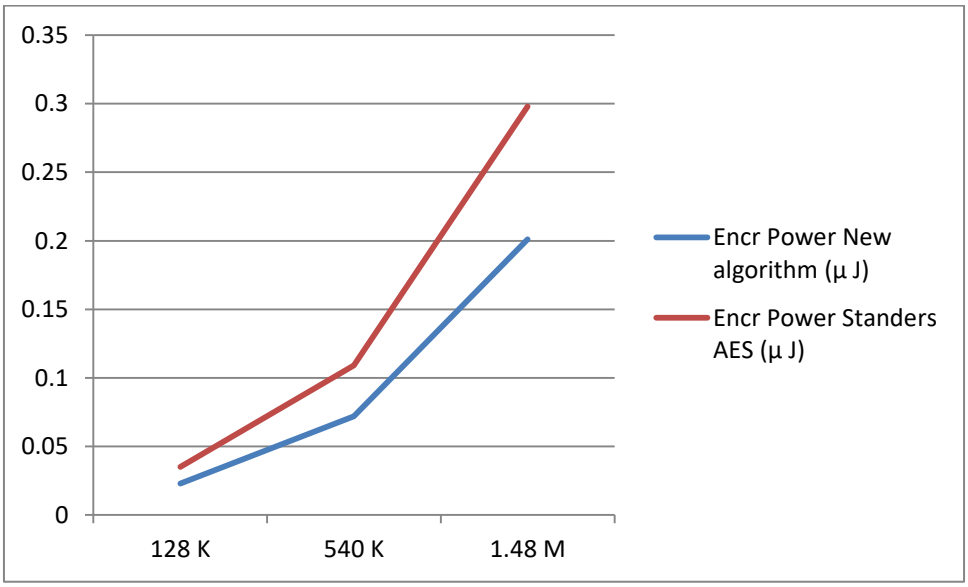


(a)



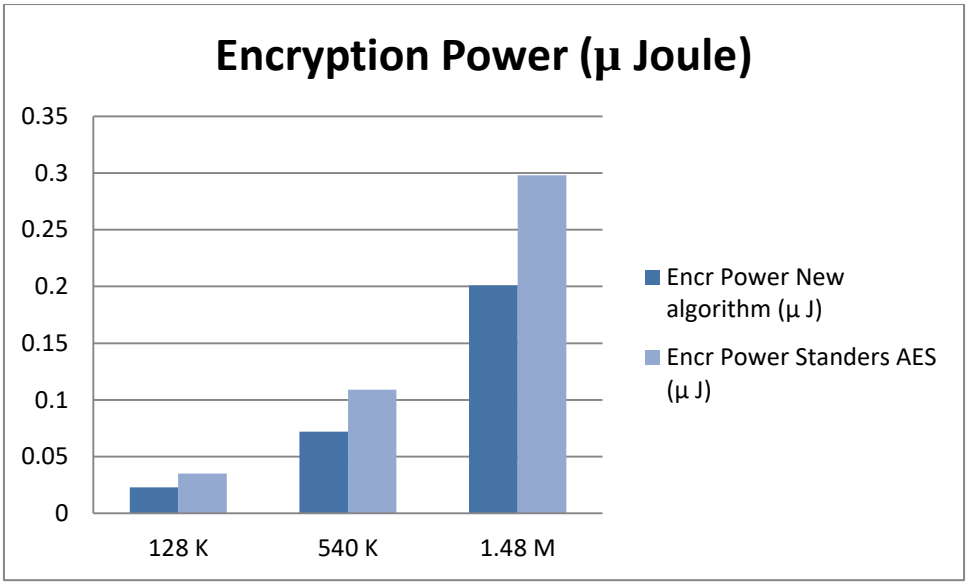
(b)

It is clear that our proposed algorithm has achieved significant time and power saving approximately 30% compared with standard AES.



(c)

Another comparison between the new TKE algorithm and standard AES has also been done with many numbers of encryption rounds to show the differences between them and their effect on the encryption cost. The new design achieved a better performance in all rounds. Table (7-13) illustrates the results for each round.



(d)

Fig. 7-11 Comparisons of AES and TKE

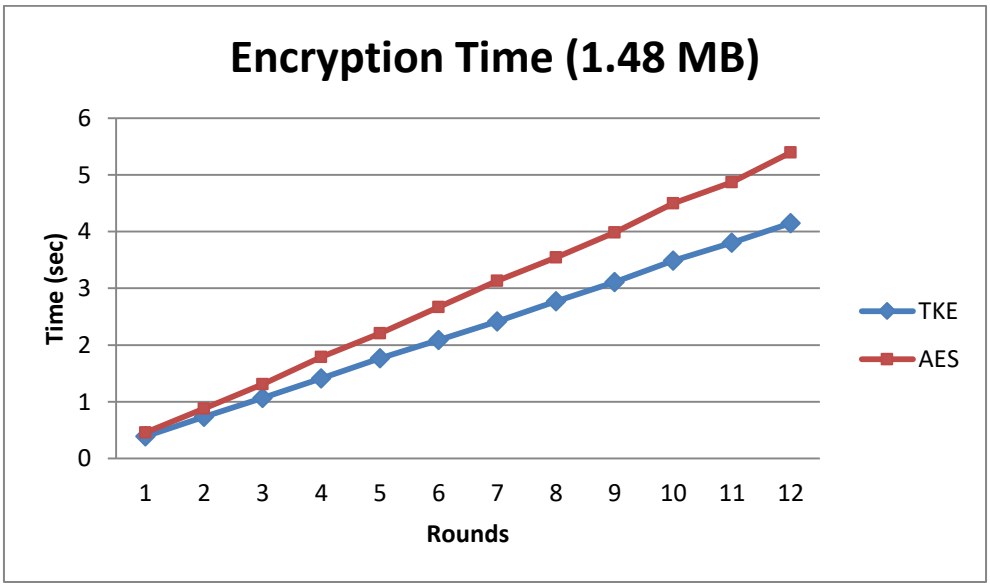
Table 7-13 Comparison for TKE and AES with many Rounds

(a) Encryption Time

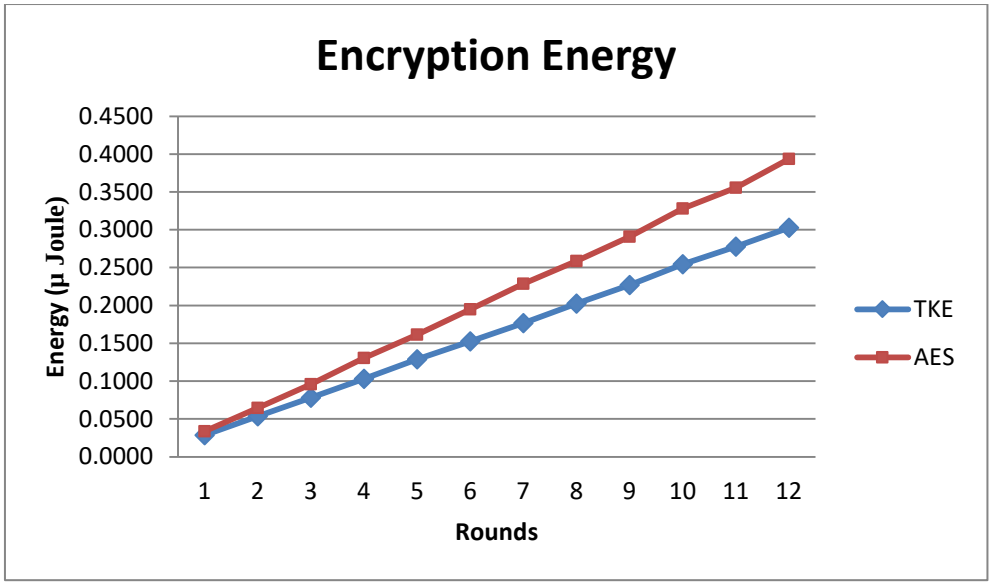
Rounds Iteration	Encryption Time TKE (Sec)	Encryption Time Standard AES (Sec)
1st Rn	0.393	0.459
2nd Rn	0.736	0.884
3rd Rn	1.07	1.313
4th	1.412	1.789
5th	1.765	2.21
6th	2.087	2.669
7th	2.417	3.131
8th	2.774	3.543
9th	3.109	3.986
10th	3.487	4.495
11th	3.804	4.872
12th	4.148	5.392

(a) Energy Consumption

Rounds Iteration	Encryption Energy TKE (µ J)	Encryption Energy Standard AES (µ J)
1st Rn	0.0287	0.0335
2nd Rn	0.0537	0.0645
3rd Rn	0.0781	0.0958
4th	0.1031	0.1306
5th	0.1288	0.1613
6th	0.1524	0.1948
7th	0.1764	0.2286
8th	0.2025	0.2586
9th	0.2270	0.2910
10th	0.2546	0.3281
11th	0.2777	0.3557
12th	0.3028	0.3936



(a)



(b)

Fig. 7-12 Comparisons of AES and TKE for Many Rounds

7.5.2 Security Analysis

The security analysis has given further consideration in this chapter to confirm the security strength of the novel design. There are three proves to confirm the security of the encrypted files that are: the human recognition of the recorded sound, the CrypTool graphs, and numerical NIST tests results. The recorded sound (cipher) after encryption was completely unclear and nobody can understand it.

The security metrics are important to measure the security strength (Ye & Huang, 2016), (Riad, et al., 2013). In chapter 3, all these metrics are explained, and in the following section explain each one with their output.

7.5.2.1 Entropy and Binary Histogram

Table (7-14) shows the Entropy test result for the encrypted file in both the AES standard and the new proposed algorithm. The new algorithm achieved a good performance compared to the standard algorithm, which was 7.99 from the maximum possible value of = 8. This means that there is significant randomness in the newly proposed algorithm, keeping the diffusion in the cipher and leading to more complexity in the relationship between the cipher and the plaintext.

Table 7-14 Entropy Analysis

Audio file	Plain	AES cipher			TKE cipher		
	Entropy	Entropy	Max. possible Entropy	Possible byte value	Entropy	Max. possible Entropy	Possible byte value
test	7.79	7.99	8	256	7.99	8	256
teaching	5.65	7.99	8	256	7.99	8	256
washing	5.4	7.99	8	256	7.99	8	256
computer	5.13	7.99	8	256	7.99	8	256

The figures below show the security analysis for the encrypted file for both the standard AES and the proposed algorithm. Fig (7-13) represents the Binary Histogram of the original audio file (computer.wav) before the encryption (Plaintext)

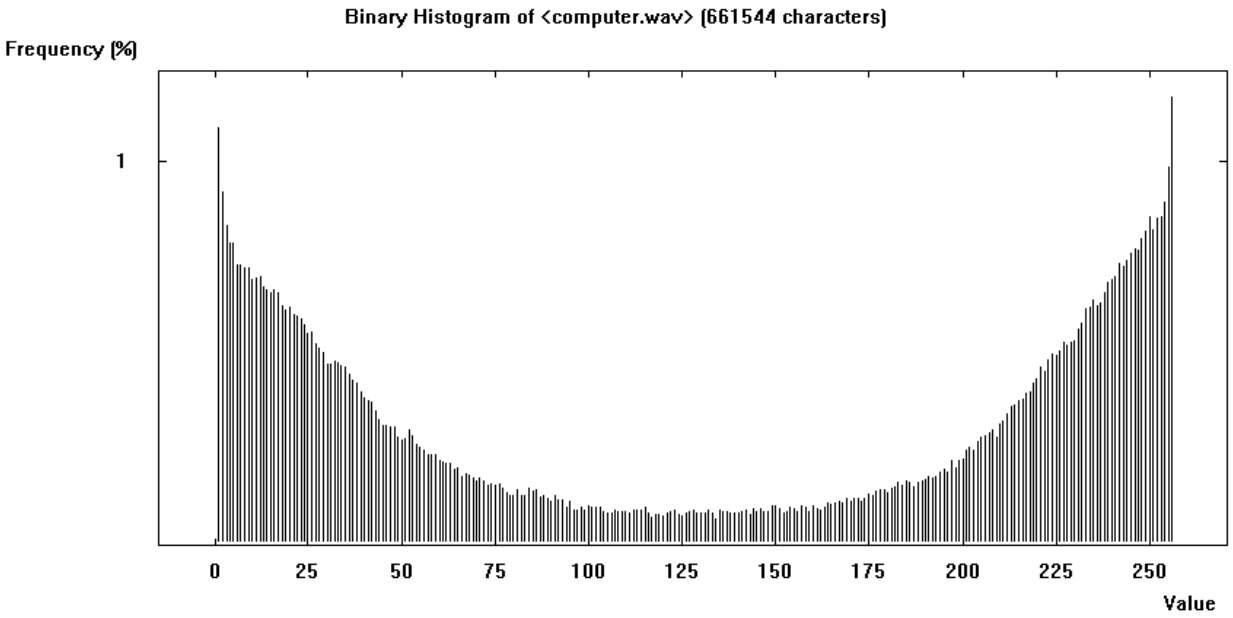


Fig. 7-13 Binary Histogram for Plain audio file

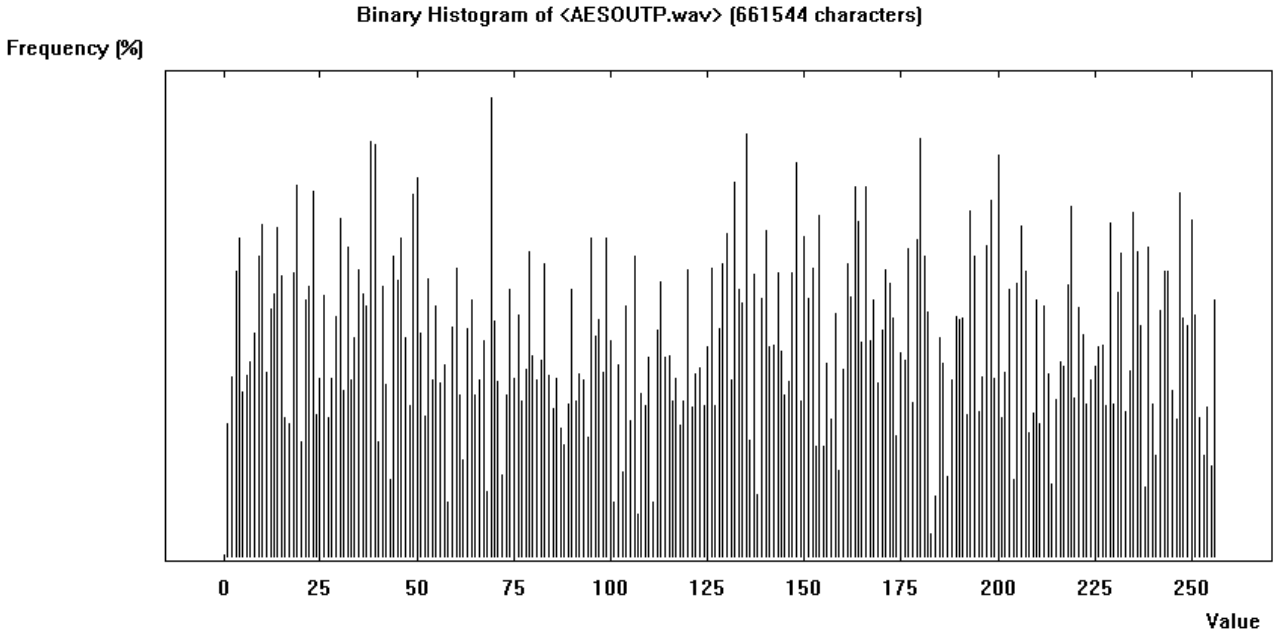


Fig. 7-14 Binary Histogram for Cipher audio file (Proposed TKE)

Fig (7-14) illustrates the Binary Histogram of the encrypted file by proposed algorithm TKE. It is clear that there is a huge difference between the original file and the encrypted file. The above figures describe the repetition percentage of each character/number in the audio file. For example, number 50 has a different value in each figure, this means that the encrypted file has good confusion pattern to trick the attacker keeping the data more secure.

Fig (7-15) shows the Binary Histogram of the encrypted file for the standard AES, in comparison with the Binary Histogram presented for the new algorithm in Fig (7-14). There is a clear similarity between the standard AES and the proposed algorithm as compared with the original file.

As illustrated in chapter 3, to understand the diagram, refer to the ASCII code. Table (7-15) can translate each number with the equivalent character. For example, (125 Dec. = 7D Hex=} ASCII) the character (125) has 0.43% frequency, While actual frequency in the plain file is 0.6%.

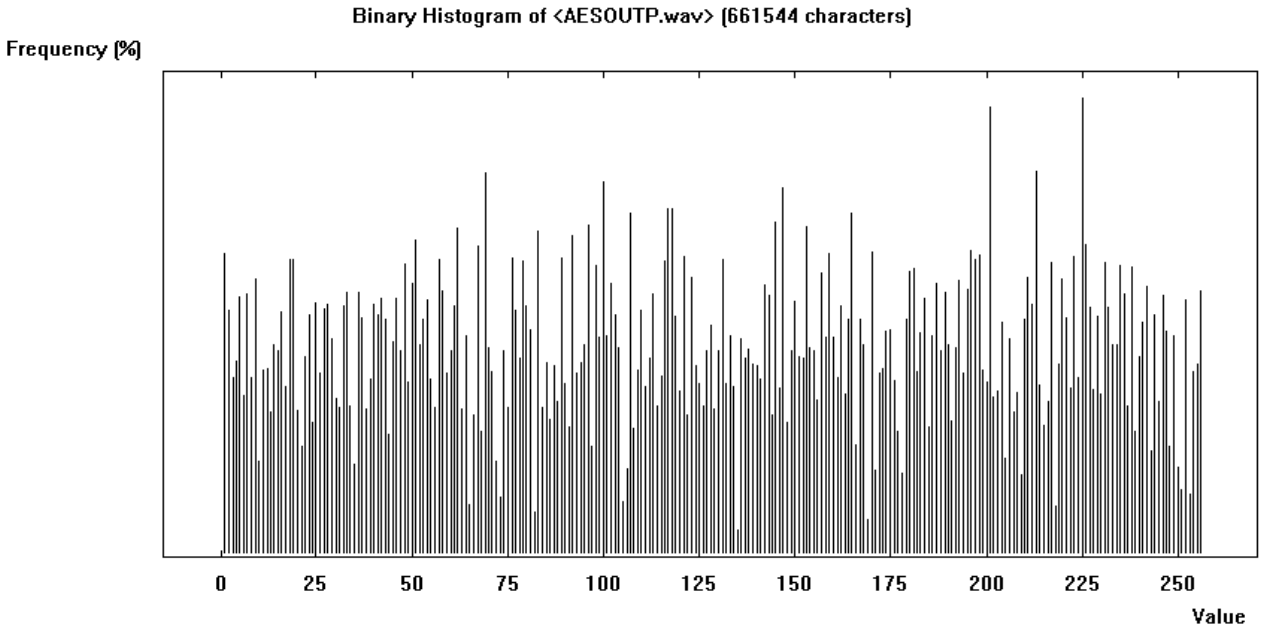


Fig. 7-15 Binary Histogram for Cipher audio file (Standard AES)

The following table shows the frequency for some characters for the plain and cipher which encrypted by standard AES and proposed TKE.

Table 7-15 Binary Histogram Values

Character value (Dec.)	Equivalent value (Hex)	Frequency (Plain Text) (%)	Frequency Cipher (TKE) (%)	Frequency Cipher (AES) (%)
1	01	**1**	0.27	0.61
25	19	**0.49**	0.37	0.51
50	32	**0.25**	0.77	0.55
75	4B	**0.13**	0.38	0.31
100	64	**0.09**	0.44	0.75
125	7D	**0.06**	0.43	0.35
126	7E	**0.061**	0.59	0.31
127	7F	**0.062**	0.31	0.41
128	80	**0.063**	0.46	0.46
150	96	**0.12**	0.63	0.52
175	AF	**0.13**	0.40	0.47
200	C8	**0.22**	0.82	0.37
225	E1	**0.46**	0.38	0.93
250	FA	**0.77**	0.67	0.18

Frequency means the number of times repeated (distributed) in the text. From the above table, it is clear that a good confusion has been achieved in the cipher for both TKE and AES

because all the frequency values have big differences from the plain file. This adding more confusion to the cipher and will confuse the attacker making it more secure and high resistance to cryptanalysis. And can be noted that TKE has a similar strength of AES. This mean the security has been maintained.

To calculate the differences between the Plain file and cipher mathematically, the following equation computes the number of each character in the file:

$$no.of\ char = Total\ no.of\ char * \frac{char\ freq.}{100}$$

For example, the number of distributed char: (125≠7D) =

$$no.of\ 7D = 661544 * \frac{0.43}{100} = 2844$$

While in the Plain file:

$$no.of\ 7D = 661544 * \frac{0.06}{100} = 397$$

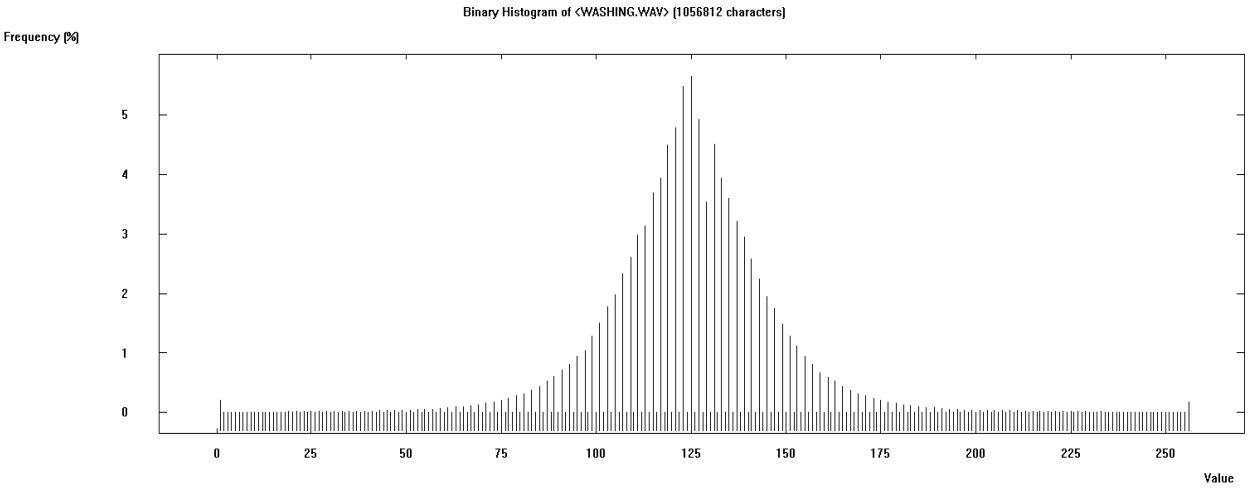
Another example of (150≠96):

$$no.of\ 96 = 661544 * \frac{0.63}{100} = 4167$$

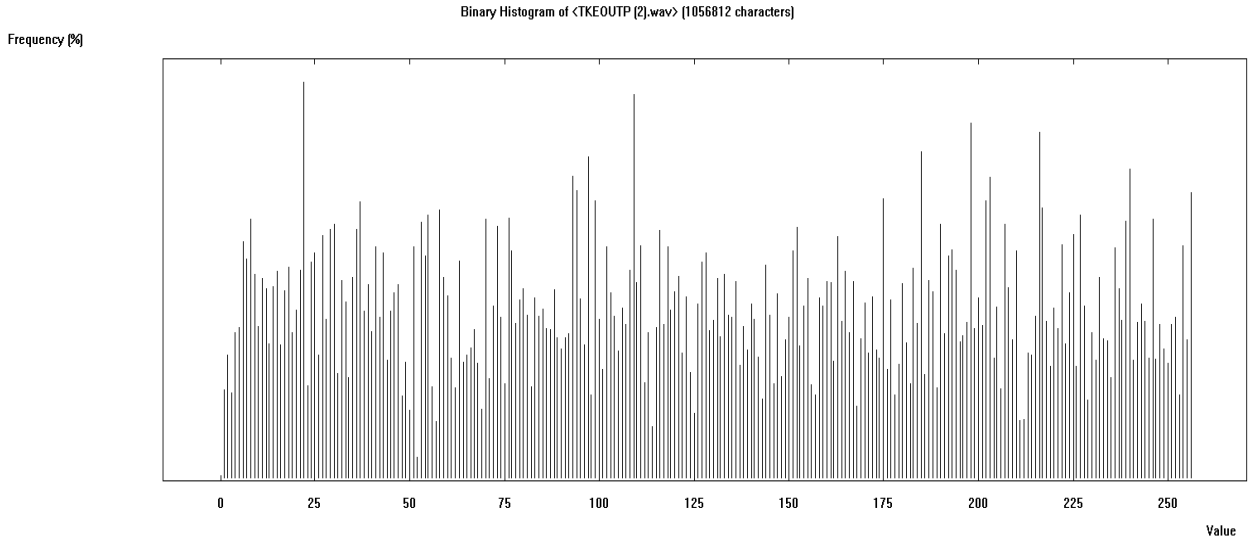
In Plain file:

$$no.of\ 96 = 661544 * \frac{0.12}{100} = 794$$

Another file (Washing.wav) has been ciphered and analysed to test its characteristics. Fig (7-16) shows the Binary Histogram for both the Plain file before the encryption and the Cipher file after the encryption.



(a) Plain



(b) Cipher

Fig. 7-16 Binary Histogram for (washing)

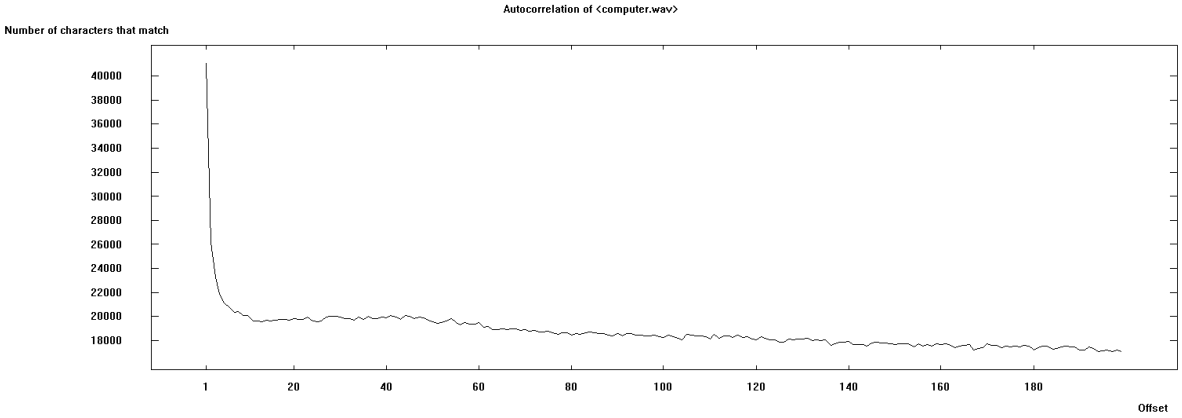
7.5.2.2 Autocorrelation

Another security parameter called Autocorrelation, (as explained in ch.3), has been considered in this analysis to show the correlation pattern for the plain and cipher. The following graphs illustrate its analysis. Table (7-16) describe the autocorrelation figs and illustrate the numerical values for each test. There is a big reduction in the correlation in the ciphers.

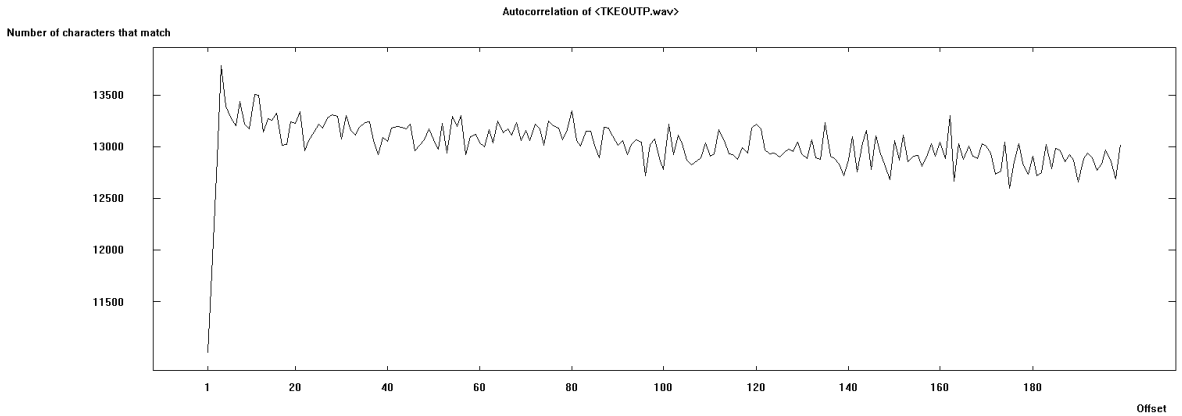
Table 7-16 Autocorrelation

Audio file	Plain		TKE		AES		Degree
	no. match	% relative agreement	no. match	relative agreement %	no. match	relative agreement %	
teaching	8000-11000	20%	4600-5100	4%	4500-5000	4%	P
Washing	48000-55000	12%	8100-8600	4%	8400-8800	4%	P
computer	19000-40000	15%	12500-13500	4%	12400-13400	4%	P

Computer

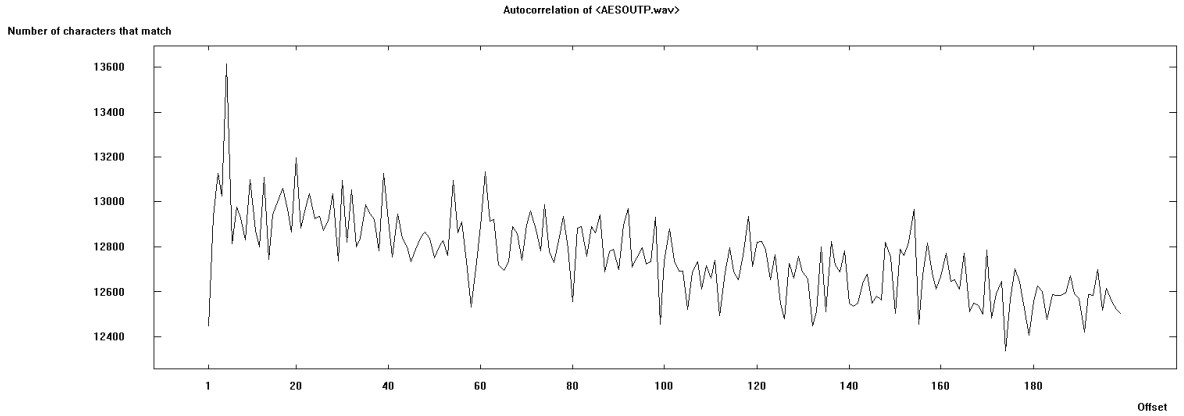


(a) Autocorrelation for Plain

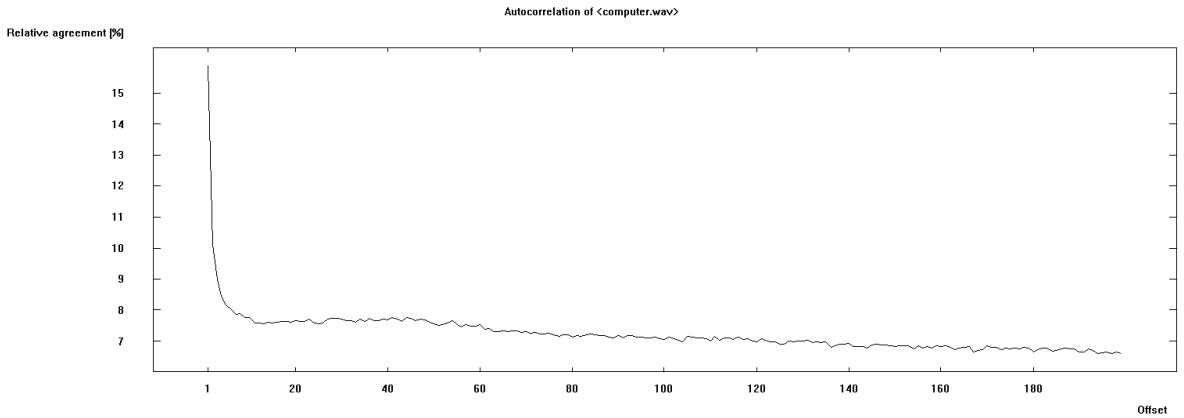


(b) Autocorrelation for Cipher by TKE

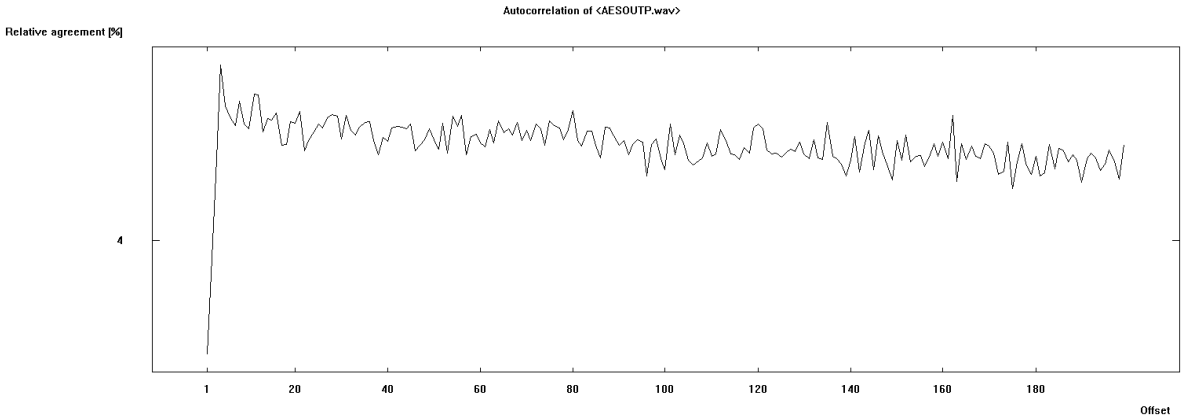
The number of characters that match is reduced in the cipher for TKE and AES and it roughly the same amount. This means that have been a similar cryptographic strength for both of them



(c) Autocorrelation for Cipher by AES

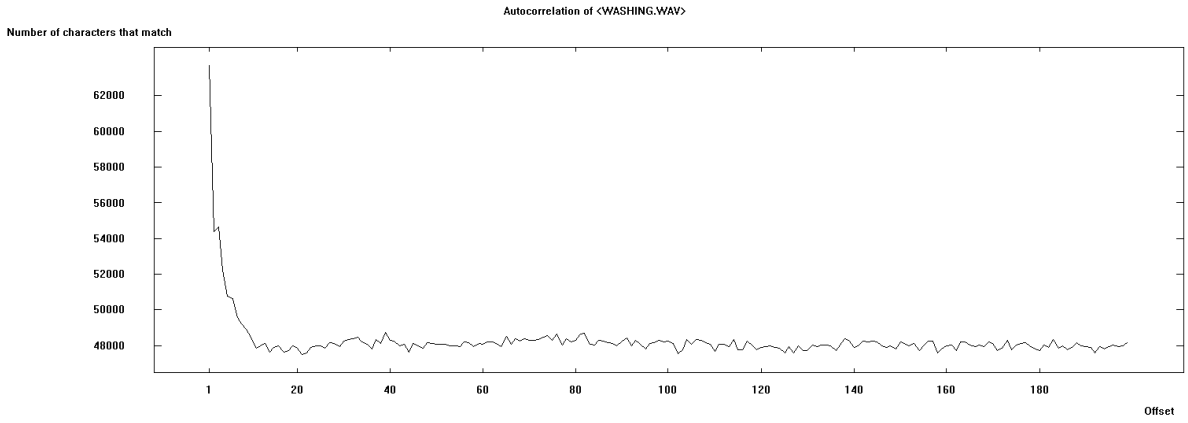


(d) Relative agreement % (Plain)

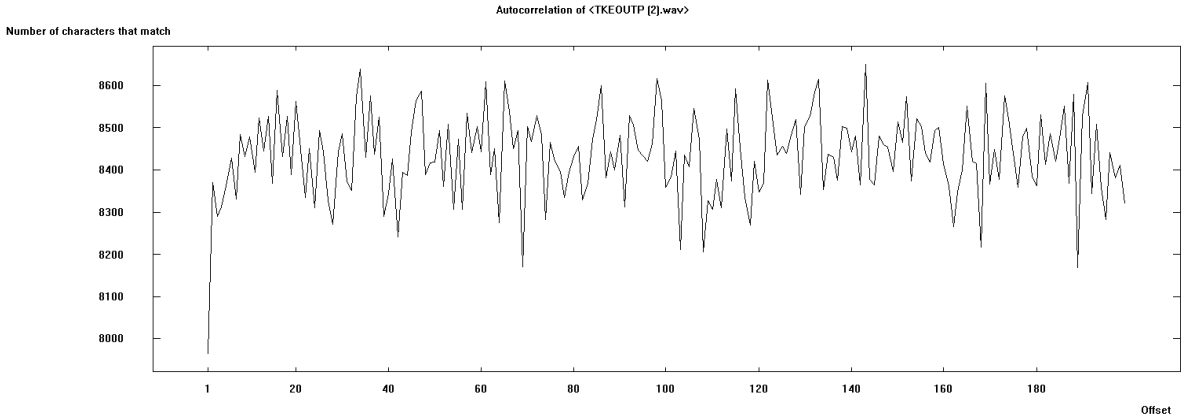


(e) Relative agreement % (Cipher)

Fig. 7-17 Autocorrelation for testing audio file



(a) Plain



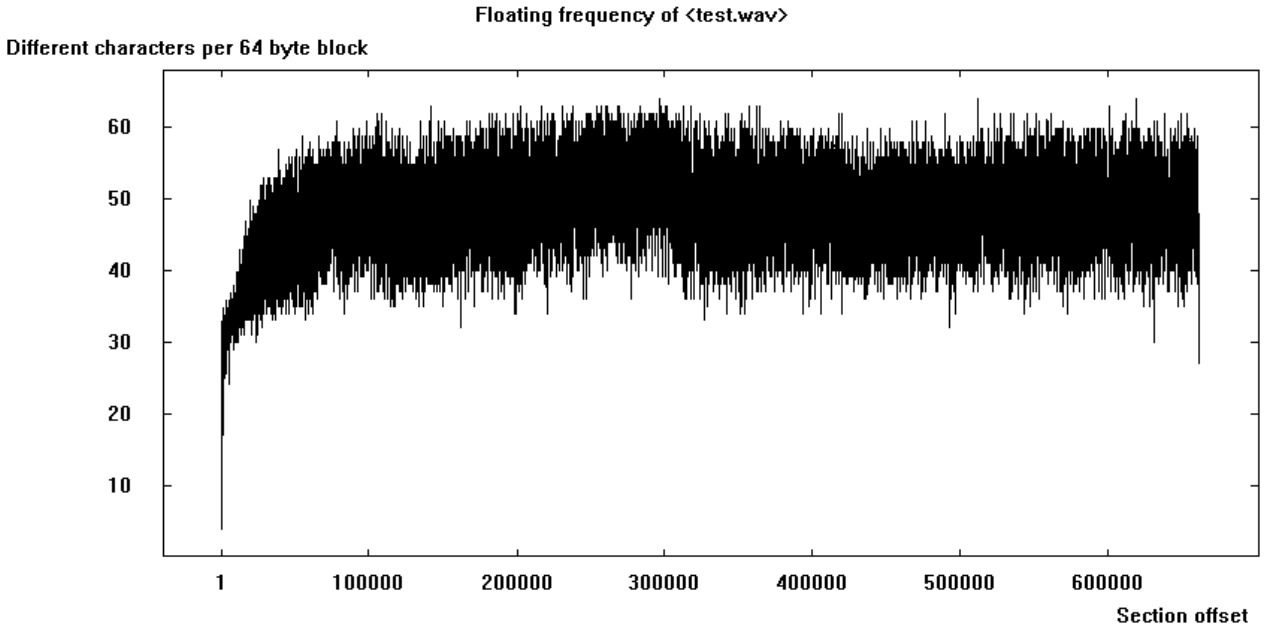
(b) Cipher

Fig. 7-18 Autocorrelation for washing file

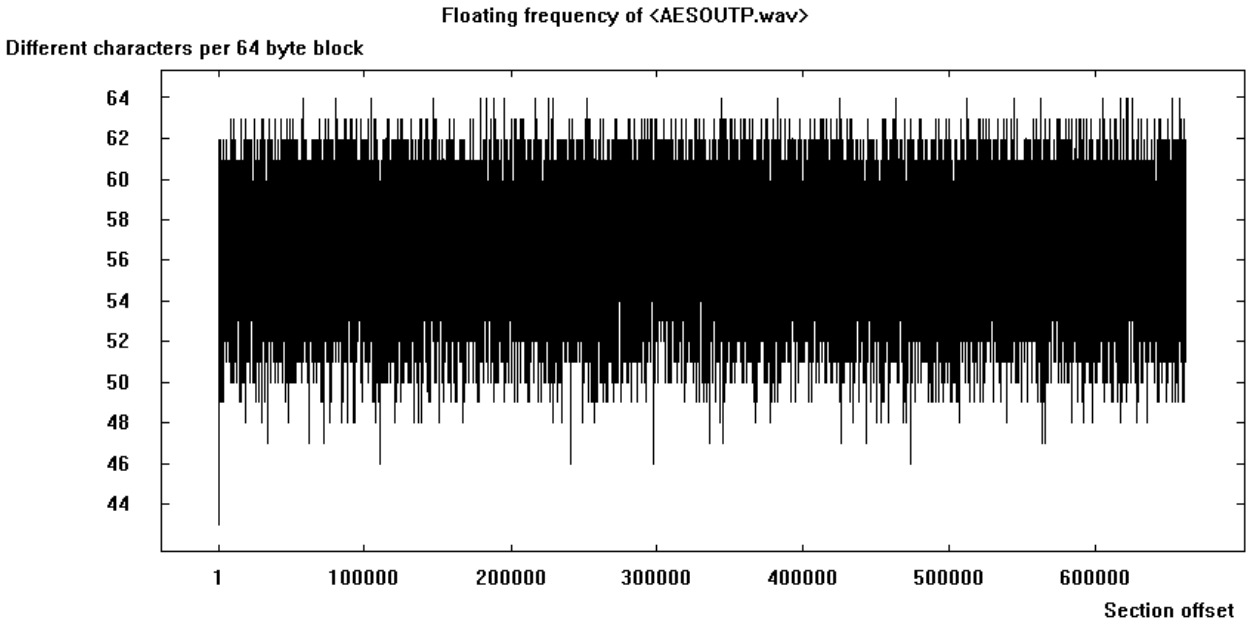
See appendix for more relative agreement %

7.5.2.3 Floating Frequency and Poker Test

Fig. (7-19) shows the Floating frequency for both the original audio file and the encrypted file for the proposed algorithm. The floating frequency describes the number of different characters per 64-byte block, higher number means higher security.



(a) Plain



(b) Cipher

Fig. 7-19 Floating Frequency

It is clear from the fig that the floating frequency for the encrypted file is good. The most of frequency is from (50 – 65) different characters per 64-byte block while it ranged from (30-55) in the plain file. This means that there is significant randomness in the newly proposed algorithm, keeping the diffusion in the cipher and leading to more complexity in the relationship between the cipher and the plaintext.

All the figures above and the analysis demonstrate that an important security level has been achieved through the newly proposed encryption algorithm. And there is reasonable behaviour between the standard AES and the proposed TKE scheme.

In addition to the previous analysis, some tests have been carried out, such as frequency test and poker test, to test the randomness of the output audio file. These tests have been carried out on the encrypted audio file and both of these tests were passed, The Poker test below also shows the significant level of randomness which has passed as shown in Fig (7-20).

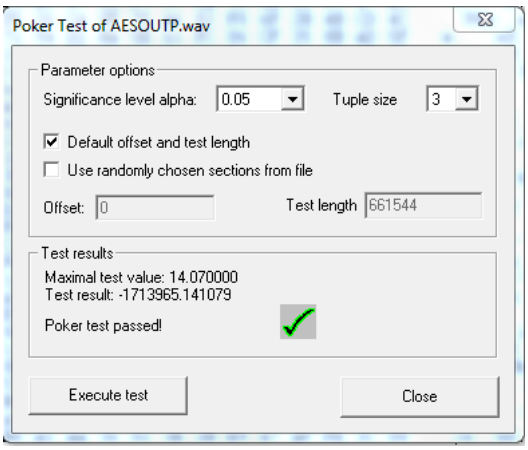


Fig. 7-20 Poker Test result

7.5.2.4 Brute Force Attack

A brute force attack is a trial-and-error method used to obtain the cryptography keys. In a brute force attack, automated software is used to generate a large number of consecutive guesses as to the value of the desired data. Brute force attacks are used by attackers to breach the encrypted data.

As explained in section (7.3) the new key space will be:

$$Triple\ Key: \quad 2^{128} * 2^{256} * 8! * 2^{128}$$

In this work, A Brute-Force attack was also tested on the 128-bit key to test the key strength. It shows that a huge time was taken to analyse the key, as in fig (7-21)

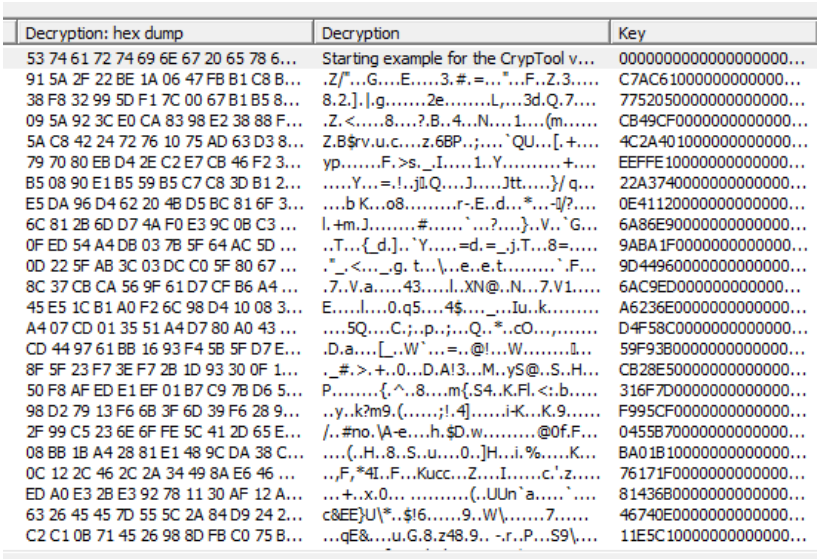


Fig. 7-21 Brute-Force attack

The table show tests result and illustrates each algorithm with their keys size and the number of possible keys (keyspace) in addition to the time required to find (recover) the encryption key.

Table 7-17 Key Reveal Time

Cipher	Key size (bit)	Keyspace	Recovery Time
RC4	32	2^{32}	5 h
AES	128	2^{128}	$2.2*10^{25}$ years
Proposed TKE	128+256	2^{384}	$9.4*10^{64}$ years

It takes billions of years to find the encrypt key. So, the brute force attack is not possible to breach this algorithm. this proves that new algorithm has a high-level of security because the complexity of finding the keys is increased, after adding more keys, each key has 2^{128} of complexity (key space), in addition of using eight S_Boxes which increased the complexity by 8!, this complexity has multiplied by the number of rounds, 10 round in standard and 9 in new AES.

7.5.2.5 Statistical Test Suite (P-RNG)

This section conducts further security analysis for the audio files before and after the encryption by the proposed TKE algorithm. As explained in ch.3, Statistical Test Suite, published by National Institute of Standard and Technology (Rukhin, et al., April 2010), is an important test to analysis the random and Pseudorandom Number Generator for cryptographic applications. Here, the frequency test has been run to test the randomness of the encrypted data. The test measure the P_value for each bit stream has been chosen by the test. Also, it counts the number of zeros and ones in each stream and calculates the result. The P_value can be calculated by using the threshold (alpha 0.01) ch.3. When the bit streams result greater than alpha, then the test is passing. To compute the P_value, the following mathematical example illustrates the steps of it:

The zeros and ones of the input sequence (*e*) are converted to values of –1 and +1 and are added together to produce S_n. For example, if

e = 1011010101, then *n*=10 and $S_n = 1 + (-1) + 1 + 1 + (-1) + 1 + (-1) + 1 + (-1) + 1 = 2$

Compute the test statistic

$S_{obs} = \frac{|Sn|}{\sqrt{n}}$, then

$S_{obs} = .632455$

P-value = $\boldsymbol{erfc}\ \frac{|Sobs|}{\sqrt{2}}$, where ***erfc*** is the complementary error function.

=

If the *P_value* ≥ 0.01, then the sequence is random. Otherwise, conclude it non-random. See appendix..

Another example can be given for 100-bit sequence,

e =1100100100001111110110101010001000100000101101000110000100011010011000100 11000110011000101000010111000

n=100 , then

S_{100}= -16

S_{abs}= 1.6

P-Value= 0.109599

Since P-value ≥0.01 , accept the sequence as random

In our test, the result can be found in the table (7-18)

Table 7-18 P_values and Test Results (Plain file)

(a) P_Value for Bit streams

Bit stream	P_value	Result	proportion
1st	0.000004	Failure	0/5
2nd	0.000063	Failure	
3rd	0.000002	Failure	
4th	0.000003	Failure	
5th	0.000002	Failure	

(b) Number of Zero's for each Bitstream

Bit S Read alpha=0.01	0s	1s
100	73	27
100	70	30
100	74	26
100	72	25
100	73	29

The above tables show the P_values and the results for each bit stream. In this test there are five-bit streams has been chosen according to our instruction.

Table 7-19 P_values and Test Results (Cipher file)

(a) Number of Zero's for each Bitstream

Bitstream	P_value	Result	proportion
1st	0.2076	Pass	4/5
2nd	0.0163	Pass	
3rd	0.1052	Pass	
4th	0.0008	Failure	
5th	0.4232	Pass	

(b) Number of Zero's for each Bitstream

Bit S Read alpha=0.01	0s	1s
100	58	42
100	54	46
100	47	53
100	62	38
100	44	56

It is clear that the cipher has a considerable randomness and 4/5 of the randomness tests have been passed. Also, the number of Zeros and ones is near the equalization.

7.5.3 Discussion

From all above analysis and tests, it is clear that the newly proposed algorithm has achieved a significant security level and it has good resistance to different attacks such as brute-force and differentiate attacks, because of the confusion and the diffusion it has. Also, the high randomness which appears in the above means that 9 rounds are still as high as possible secure. Also, the QoS parameters have been improved to reach 30% compared with the standard AES.

The results show that significant improvements in the proposed algorithm were achieved. The significance of the results is measured using a T-test function. The critical value for this test is (0.05). The output of the test shows that the *P-value < 0.05.*

In the quality test: $P\text{-value} = 0.000387 < 0.05$

For security test: $P\text{-value} = 0.000119 < 0.05$

So these results are considered as significant and the algorithm has achieved a good tradeoff in security and quality.

From the proposed schemes in chapter 6 and 7, there are three main algorithms have been proposed. Each one has a specific characteristic in term of latency, Energy, and complexity. Table (7-20) illustrates and compares each algorithm

Table 7-20 Three algorithm comparison

Property	Proposed L-Mixcol	Proposed LEA L-9 round	Proposed TKE
S_box	Multi	Multi	Multi
Num. of keys	2	2	3
Key length	128 bit	same	same
Numbers of rounds	10	9	9
Security (key space)	$2^{128}*2^{256}*8!$	$2^{128}*2^{256}*8!$	$2^{128}*2^{256}*8!*2^{128}$
Single round details	Four function	same	same
Block length	16 Bytes	same	same
Encryption time	**0.78 T**	**0.65 T**	**0.7 T**
Power consumption	**0.78 P**	**0.65 P**	**0.7 P**
Energy Saving %	**22%**	**35%**	**30%**
Output file size = input file size	Yes	Yes	Yes

The best chose is TKE algorithm, as it makes a significant tradeoff solution (balance) between Energy and Security. Table (7-21) shows the comparison of features between standard AES and proposed TKE algorithm. It can be seen that there are many important differences between them. However, it is clear that the file size is still the same, which will therefore not affect the memory size and the bandwidth of network paths. As mentioned in ch.3, there are ranges of encryption algorithms can be designed but it should consider the security strength. Therefore the best choose in this research is the TKE and LEA because it achieved the balance between the security and the energy saving.

Table 7-21 TKE ans AES features comparison

Property	Standard AES	Proposed TKE
S_box	Single	Multi
Num. of keys	1	3
Key length	128 bit	same
Numbers of rounds	10	9
Security (key space)	2^{128}	$2^{128}*2^{256}*8!*2^{128}$
Binary Histogram	Random	Random
Autocorrelation	Low	Low
Poker Test	Pass	Pass
Single round details	Four function	same
Block length	16 Bytes	same
Encryption time	**T**	**0.7 T**
Power consumption	**P**	**0.7 P**
Energy Saving %	**0%**	**30%**
Output file size = input file size	Yes	Yes

From the above table it is clearly that the proposed scheme has the same strength as in AES according the security parameters in the table. Also, the new algorithm has a higher-level of security because the complexity of finding the keys is increased, after adding more keys, each key has 2^{128} of complexity, as mentioned in table (7-21) in addition of using eight S_Boxes which increased the complexity by 8!, this complexity has multiplied by the number of rounds, 10 rounds in standard and 9 rounds in new AES. Meanwhile, the execution time and power consumption have decreased by 30% and this is because of reducing the rounds and new mixcolumn function. Therefore, the new algorithm is better than AES.

The new Key space:

$$Triple\ Key: \quad 2^{128} * 2^{256} * 8! * 2^{128}$$

Figure (7-22) explain the final proposed algorithm chart and the standard AES. There is a clear difference between both of the two algorithms.

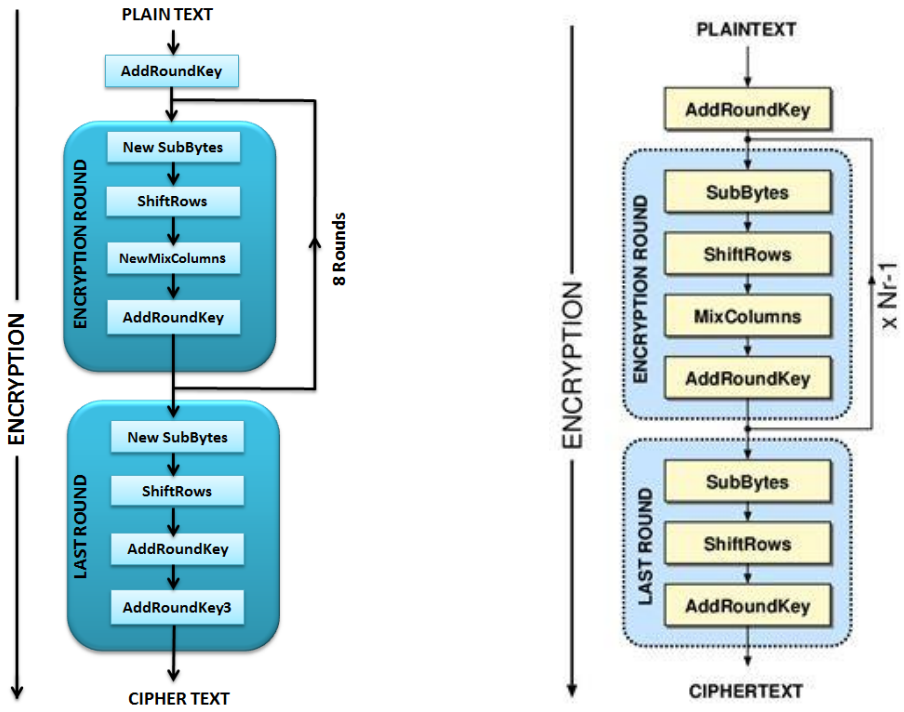


Fig. 7-22 Two Design Comparison

Finally, a comparison with previous studies has been conducted to validate the proposed work and confirm the research objectives. Table (7-22) show the list of published work and our work and illustrate the power saving percentage and the security level for each one.

Table 7-22

Work ref.	Power saving (%)	Overall Security against AES
(Msolli, et al.,2016)	40 %	low
(Ali, et al., 2014)	0 %	high
This research	30 %	Same/high

It is clear that this research has achieved 30% power saving and keeping the security strength. It is better than previous work which focuses on one solution and neglects the other. This means a significant tradeoff solution (balance) between security and QoS metrics has been achieved in the proposed algorithm, making it more suitable for wireless devices.

7.6 Summary

A Triple Key encryption algorithm for audio files was developed and tested in this chapter. The main objectives have been achieved by reducing the execution time and energy consumption of the encryption process, in addition, to increasing the security level by using the third key, which not affect the encryption performance.

This chapter combines all the developed functions proposed in this research. The proposed functions are, Low computation Mixcolumn, Nine rounds iteration, and Third Key encryption. The testing and experiments were conducted and a range of implementation scenarios are setup with different audio files. Also, the algorithm has been tested with many iteration rounds to test their performance and effect. The test results show significant improvements in new design metrics. The comparison between the new algorithm TKE and the standard AES/published work shows a significant amount of time and energy consumption reduction being achieved (approximately 30%) with a higher security level.

Data security was analyzed using NIST testing tools, to measure the new algorithm strength. Many security parameters have been tested such as binary histogram, autocorrelation, and others to test the randomness and the complexity of the cipher.

The validation and evaluation showed that a significant level of security has been achieved by a new algorithm making it able to resists different types of known attacks. In addition to the significant performance, (according to the T-test), make it effective and energy saver. The algorithm TKE has achieved a considerable tradeoff improvement in security and quality. The new design is more suitable for the wireless environment and helps to address the wireless devices limitation and security concerns. By the end of this chapter, the main aim of the research has been achieved and all the objectives already addressed.

8 Conclusion & Future Research

A new encryption schemes for real time traffic over wireless networks is proposed, tested and validated in this research. Using robust changes in the standard AES functions, the proposed algorithms can provide a reasonable level of security and meets the QoS of such networks and devices. The change in the Sub-Byte function is to the increase the degree of complexity within the same delay time during the encryption and decryption processes. The change in the MixColumn function reduces the number of operations needed to multiply two matrices as it reduces the summation and multiplication process, consequently leading to the reduction of the time and energy consumption needed for this function. Finally, the proposed 9-round change is capable to decrease the power consumption by up to 10% and reduce the execution time by nearly the same proportion in addition to involving a 3rd key for encryption. The overall design helps to save (30-35%) of energy during the processing time, which make the schemes lightweight and energy saver. The main contribution of the research is reducing the execution time and power consumption of the proposed new algorithms, besides, maintaining/increasing the security level comparing with the standard AES algorithm. A comprehensive security analysis has been conducted to test the validity of the proposed algorithms in terms of the high complexity and randomness of the proposed algorithm which can resist different types of attacks. The proposed algorithm is suitable for wireless devices with limited resources and it achieves a considerable trade-off solution between security and QoS, thus it exhibits its applicability for any wireless networks where the resources are limited.

The aim of this research was to *find an efficient security system for encrypting the voice and multimedia information data without the expensive cost*, and it has been achieved. The main research question, which was *"How can the security cost, (delay, energy), be reduced without affecting the security level of cryptography implementation for voice over wireless devices?"* has been answered. This achievement has been achieved by reducing the execution time and power consumption of the proposed new Lightweight cryptography scheme, besides, maintaining/increasing the security level comparing with the standard AES algorithm.

Therefore, this research has contributed to the knowledge by addressing the following objectives:

- Objective #1 was a critical investigation into the wireless network traffic and standard cryptography algorithms.
- Objective #2 was investigating and studying the network traffic and their requirements. Also, it experimentally investigates the AES encryption algorithm.

Chapter 2 and 4 have shown that Wireless networks are the most important part of IoT, and their security is crucial. They also shown that the encryption is the main principle of security because it keeps the confidently of the information and the best encryption algorithm is AES algorithm because of its strength; however this algorithm needs more research to meet the QoS requirements of the new wireless devices such as the energy of these devices and the time (delay) for voice traffic. Furthermore, these chapters showed that Real-time applications and voice are very important in wireless networks. Therefore, securing these applications and keeping their quality is a big challenge. These chapters explained that the previous works could not solve the problems in the voice delay and in the power of wireless devices in addition to the security level of algorithms.

This phase helps to propose the best solution for the current gap, by a lightweight encryption algorithm to be suitable for V-over-WMANET and helps to get a good tradeoff between the security and QoS metrics and don't pass the security threshold.

In addition to the literature review, this chapter gave a clear image to understand the network requirement and their behavior. The importance of the knowledge of characteristics of each type is useful because it will help us to understand the QoS metrics for each type of traffic. Also, this chapter showed the execution performance of the standard AES algorithm and its strength.

The investigation of the standard AES algorithm showed its performance, in term of execution time and energy consumption, in addition to the security analysis of the encrypted output. These investigations will help to compare it with the proposed schemes to validate and evaluate their results. So the objective of this chapter has been achieved in a good manner.

AES is very secure because it used substitution, permutation, mixing, and keys, in addition to many rounds iteration. These operations offer the confusion and diffusion needed to protect any cipher from cryptanalysis attempt. So, any proposed cryptosystem based on AES features will be efficient and secure. In this research, our proposed crypto-schemes based on AES features because of its strength.

- Objective #3 was the conceptual development of an innovative encryption algorithm with a high level of complexity and at the same time keeping the execution time and power consumption at the same level. And Address the fixed structure of SubByte transformation function, to increase the confusion in the cipher.

Chapter 5 has developed A SubByte function using multi S-box transformation technique to increase the confusion and complexity of the encryption algorithm. The complexity of the proposed algorithm has increased and the time and energy consumption kept in the same amount. The output of this chapter has been used to develop the new lightweight algorithms, proposed in other chapters, and it is the base for proposed functions. Another SubByte transformation functions have been suggested in this chapter, however, they can be used in future research. So, chapter 5 proposed three S-box generation methods but just one has been chosen to be involved in the new design.

- Objective #4 to design, implement and build a lightweight and low-Energy encryption algorithm for audio files considering the cryptosystem strength. To meet the wireless devices requirements (limitation)

A lightweight and low energy encryption algorithm for audio files was developed and tested in chapter 6. The main objectives have been achieved by reducing the execution time and energy consumption of the encryption process compared with the standard algorithm (AES) and keeping its security level in a good complexity.

A Low computation Mixcolumn function and nine rounds iteration for the new algorithm LEA have been proposed in this chapter. The testing and experiments were conducted and a range of implementation scenarios are setup with different audio files. Also, the algorithm has been tested with many iteration rounds to test their performance and effect. The test results show significant improvements in new design metrics. The comparison between the new algorithm and the standard one shows a significant amount of time and energy consumption reduction being achieved (approximately 35%)

Data security was analyzed using specific testing tools, to measure the new algorithm strength. Many security parameters have been tested such as binary histogram, autocorrelation, and others to test the randomness and the complexity of the cipher.

The validation and evaluation showed a significant level of security has been achieved by the new algorithm. In addition to a good performance makes it cost-effective and energy saver. The algorithm LEA has achieved a good tradeoff in security and quality. The new design is

more suitable for the wireless environment and helps to address the limitation of the wireless devices.

- Objective #5 to propose a novel encryption algorithm with triple key and high level of complexity with effective execution costs such as time and energy consumption. Also, to Validate, Evaluate and Analyses the new security architecture in every step of the design to prove their strength.

Chapter 7 has developed and tested A Triple Key encryption algorithm for audio files. The main objectives have been achieved by reducing the execution time and energy consumption of the encryption process, in addition, to increasing the security level by using the third key, which not affect the encryption performance.

This chapter combines all the developed functions proposed in this research. The proposed functions are, Low computation Mixcolumn, Nine rounds iteration, and Third Key encryption. The testing and experiments were conducted and a range of implementation scenarios are setup with different audio files. Also, the algorithm has been tested with many iteration rounds to test their performance and effect. The test results show significant improvements in new design metrics. The comparison between the new algorithm TKE and the standard AES/published work shows a significant amount of time and energy consumption reduction being achieved (approximately 30%) with a higher security level.

Data security was analyzed using NIST testing tools, to measure the new algorithm strength. Many security parameters have been tested such as binary histogram, autocorrelation, and others to test the randomness and the complexity of the cipher.

The validation and evaluation showed a significant level of security has been achieved by a new algorithm making it able to resists different types of known attacks. In addition to a good performance make it cost-effective and energy saver. The algorithm TKE has achieved a considerable tradeoff in security and quality. The new design is more suitable for the wireless environment and helps to address the wireless devices limitation and security concerns.

To conclude the whole research, there are more than 120 tests, for both the performance measurement and security analysis, have been conducted in this research. The results have been carefully validated and evaluated and it shows significant improvements in new design metrics for proposing a lightweight encryption algorithm with the same level of security compare with the standard algorithm.

The main contribution of the research is reducing the execution time and power consumption of the proposed new algorithm, besides, maintaining/increasing the security level comparing with the standard AES algorithm.

A comprehensive security analysis is conducted to test the validity of the proposed algorithm in terms of the high complexity and randomness of the proposed algorithm which can resist different types of attacks, so, the scheme is absolutely secure.

The proposed algorithm is suitable for wireless devices with limited resources and it achieves a considerable trade-off solution between security and QoS, thus it exhibits its applicability for any wireless networks where the resources are limited.

The impact of this work is significant because it addressed both the security concerns and the implementation cost together, and achieved a reasonable trade-off for the wireless environment; the proposed scheme has balanced the security and performance by reducing the cost of the security without affecting security level or passing the threshold.

The importance of this work comes with the hugely needs of security in next year 5G revolution (2020). The small sensor nodes (with limited battery power) are continuously required a lightweight security implementation to do their function. This work should promise to offer a range of solutions to the networks designer to agree to their requirements.

Future Research:

Future Further development of the algorithm can be performed in the future, such as increasing the SybeByte function complexity or involving or adding the third key in the algorithm to increase the security level, depending on the resources and limitations of the network.

Another technique could be proposed in the future by using two S-boxes to increase the security level. The idea is to XOR two S-boxes and uses the new S-box in the transformation process for SubByte function. This method needs just one K and C to generate addition S-box and then XOR it with the existing one. This method will help to reduce the key generation size and generate just one S-box. So there is no need to generate multi S-boxes.

A future possibility to a proposed new function called Key XOR S-box method. The idea of this method is to XOR exist S-box with a chosen key, to increase the complexity of the

algorithm. So, no need to generate multi S-boxes. The method will also be cost-effective and do not affect the execution time.

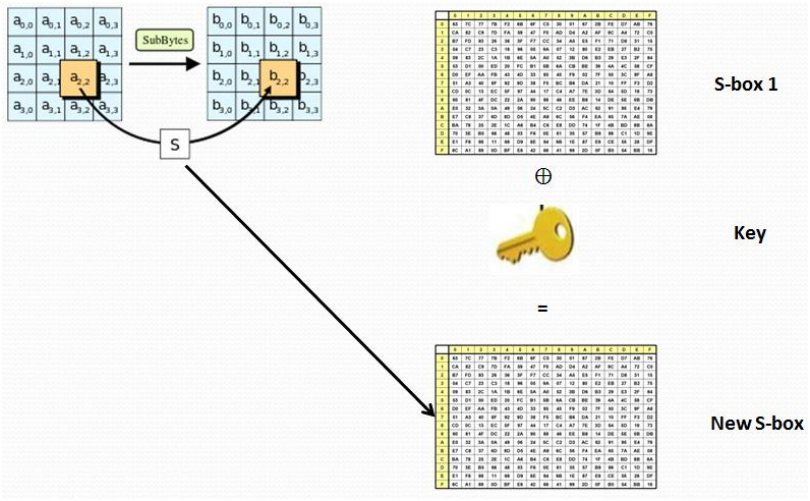


Fig 9-1 the future work

Furthermore, the implementation of the new algorithms in the real scenario could be tested in such as mobile network and this can be as cooperation with the communication companies. By the end of this chapter, the whole research work has been done and all the research objectives have been achieved.

References

Acharyya, I. S., Ezdiani, S., Sivakumar, S. & Al-Anbuky, A., 2017. Wireless Sensor Network Softwarization: Towards WSN Adaptive QoS. *IEEE Internet of Things Journal,* 4(5), pp. 1517 - 1527.

Aarti & Tyagi, S. S., 2013. Study of MANET: Characteristics, Challenges, Application and Security Attacks. *International Journal of Advanced Research in Computer Science and Software Engineering,* 3(5), pp. 252-257.

Abhijith, P., Srivastava, M. & Mishra, A., 2013. *High performance hardware implementation of AES using minimal resources.* India, International Conference on Intelligent Systems and Signal Processing (ISSP).

Abhiram, L. S., Sriroop , B. K. & Punith.Kumar , H. L., 2015. *FPGA implementation of dual key based AES encryption with key Based S-Box generation.* India, IEEE, 2nd International Conference on Computing for Sustainable Global Development (INDIACom).

Ahmad, A., Swidan , A. & Saifan , R., 2016. Comparative Analysis of Different Encryption Techniques in Mobile Ad Hoc Networks (MANETS). *International Journal of Computer Networks & Communications (IJCNC),* 8(2), pp. 89-101.

Alamsyah, Bejo, A. & Adji, T., 2017. *AES S-box construction using different irreducible polynomial and constant 8-bit vector.* Taiwan, Dependable and Secure Computing, IEEE Conference.

Albonda, H., Tapaswi, S., Yousef, S. & Cole, M., March 2017. The impact of mobility and node capacity on voice traffic. *International Journal of System Assurance Engineering and Management,* 8(33), pp. 1 - 9.

Ali, N. M., Rahma, A. S. & Jaber, A. M., 2014. Random Key Permutation Stream Algorithm Based on Modified Functions in AES Algorithm. *International Journal of Engineering & Technolog,* 4(6), pp. 367-373.

Ali, N., Rahma, A., Yousef, S. & Jaber, M., June 2014. Random Key Permutation Stream Algorithm Based on Modified Functions in AES Algorithm. *International Journal of Engineering and Technology,* 4(6), pp. 367 - 373.

Alsalami, Y., Yeun, C. Y. & Martin, T., 2016. *Linear and differential cryptanalysis of small-sized random (n, m)-S-boxes.* Spain, IEEE, 11th International Conference for Internet Technology and Secured Transactions (ICITST).

Alyasseri, Al-Attar, Z. & Ismail, N., 2014. Parallelize Bubble Sort Algorithm Using OpenMP. *International Journal of Advanced Research in Computer Science and Software Engineering,* 4(1).

Amine, M., Maglaras, L. & Argy, A., 2018. Security for 4G and 5G cellular networks: A survey of existing authentication and privacy-preserving schemes. *Elsevier-Journal of Network and Computer Applications,* Volume 101, pp. 55-82.

Apietro , R. D., Guarino, S. & Verde, 2014. Security in wireless ad-hoc networks. *Elsevier- Computer Communications,* Volume 51, pp. 1-20 [available at ScienceDirect].

ARB, 2011. *OpenMP 3.1,* s.l.: OpenMP. [online]: https://www.openmp.org/press-release/openmp-3-1-released/.

Ayyappadas, Devassy, George, S. & Devassy, A., 2014. Survey of Symmetric Cryptographic Algorithms. *Journal of Electronics and Communication Engineering (IOSR-JECE),* pp. 65-75.

Baas , B. & Liu, B., 2013. Parallel AES Encryption Engines for Many-Core Processor Arrays. *IEEE Transactions on Computers,* 62(3), pp. 536-547.

Bahrak , B. & Aref, M. R., July 2008. Impossible differential attack on seven-round AES-128. *IET Information Security, Institution of Engineering and Technology,* 2(2), pp. 28-32. [avalible online on]: https://ieeexplore.ieee.org/document/4558840/.

Balagurusamy, E., 2009. *FUNDAMENTALS OF COMPUTERS.* India: Mc Graw Hill.

Bansod, G., Raval, N. & Pisharo, N., 2015. Implementation of a New Lightweight Encryption Design for Embedded Security. *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY,* 10(1), pp. 142-151.

Barnes , A., Fernando, R., Mettananda, K. & Ragel, R., 2012. *Improving the throughput of the AES algorithm with multicore processors.* India, IEEE 7th International Conference on Industrial and Information Systems (ICIIS).

Barney, B., 2018. *Introduction to Parallel Computing,* USA: U.S. Department of Energy/ Lawrence Livermore National Laboratory / Contract DE-AC52-07NA27344, [online]: https://computing.llnl.gov/tutorials/parallel_comp.

Basaras , P., Belikaidis, I. & Maglaras, L., 2016. *Blocking epidemic propagation in vehicular networks.* Italy, 12th Annual Conference on Wireless On-demand Network Systems and Services (WONS).

Bhalshankar, S. & Gulve, A., 2015. Audio Steganography: LSB Technique Using a Pyramid Structure and Range of Bytes. *International Journal of Advanced Computer Research,* 5(20).

Bhaskar , S. M. & Ahson, S. I., 2008. *Information Security: A Practical Approach.* 1 ed. s.l.:Alpha Science International Ltd, [ISBN-10: 1842654586]..

Binod & Hyuk, 2010. Secure Multimedia Streaming over Multipath Wireless Ad hoc Network. *Research gate.*

Blaze, M., Loannidis, J. & Keromytis, A., 2001. *Trust Management for IPsec,* US: Columbia University,.

Bogdanov, A., Knudsen, L. R., Leander, G. & Paar, C., 2007. PRESENT: An Ultra-Lightweight Block Cipher. In: *Cryptographic Hardware and Embedded Systems - CHES.* Berlin-Germany: Springer-International Workshop on Cryptographic Hardware and Embedded Systems, pp. 450-466.

Brahim, M. B. & Mir, Z., 2017. QoS-Aware Video Transmission Over Hybrid Wireless Network for Connected Vehicles. *Resource Management in Vehicular Ad-Hoc Networks: Energy Management, Communication Protocol and Future Applications, IEEE Access,* Volume 5, pp. 8313 - 8323.

Bucholz, 2001. *Advanced Encryption Standard,* s.l.: s.n.

Butt, J. et al., 2018. A desktop 3D printer with dual extruders to produce customised electronic circuitry. *Frontiers of Mechanical Engineering,* 13(4), p. 528–534.

Chandramouli, R., Bapatla, S. & Subbalakshmi, K. P., 2006. Battery power-aware Encryption. *ACM Transactions in Information and System Security,* 9(2), pp. 162-180.

Chapman, B., Jost, G. & Pas, R., 2008. *Using OpenMP Portable Shared Memory Parallel Programming.* London, England: The MIT Press Cambridge, Massachusetts.

Cheng, s., Chen, P. & Lin, C., 2017. Traffic-Aware Patching for Cyber Security in Mobile IoT. *IEEE Communications Magazine,* 55(7), pp. 29 - 35.

Chouhan, T. S., 2016. *Implementation of Present Cryptographical Algorithm for the Encryption of Messages in NETFPGA 1G.* India, IEEE International Conference on Computational Intelligence and Communication Networks (CICN).

Choy, L. T., 2014. The Strengths and Weaknesses of Research Methodology: Comparison and Complimentary between Qualitative and Quantitative Approaches. *IOSR Journal Of Humanities And Social Science (IOSR-JHSS),* 19(4), pp. 99-104.

Cisco, 2016. *What is a wireless network vs. a wired network?,* US: Cisco Network academy - https://www.cisco.com/c/en/us/solutions/small-business/resource-center/work-anywhere/wireless-network.html.

Cisco, 2018. *What Is Cybersecurity?,* US: Cisco Network Academy.

Comado, n.d. *What is Man-in-the-Middle Attack?,* s.l.: Secure Box, online: https://securebox.comodo.com/ssl-sniffing/man-in-the-middle-attack/.

Conti, M., Mancini , L. V., Riccardo, S. & Verde, N., 2016. Analyzing Android Encrypted Network Traffic to Identify User Actions. *IEEE Transactions on Information Forensics and Security,* 11(1), pp. 114-124.

Daemen, J. & Rijmen, V., 2003. *AES Proposal: Rijndael,* US: National Institute of Standard.

Daernen , J. & Rijrnen, V., 2002. *The Design of Rijndael.* Germany, Springer.

Das , I., Nath , S., Roy , S. & Mondal, S., 2013. *Random S-Box generation in AES by changing irreducible polynomial.* India, International Conference on Communications, Devices and Intelligent Systems (CODIS).

Dworkin, M. J. et al., November 2001. *Advanced Encryption Standard (AES),* USA: National Institute for Standard and Technology (NIST).

Ebrahim, 2013. Symmetric Algorithm Survey: A Comparative Analysis. *International Journal of Computer Applications,* 61(20), pp. 12-19.

Ehsan , S. & Hamdaoui, B., Jun 2012. A Survey on Energy-Efficient Routing Techniques with QoS Assurances for Wireless Multimedia Sensor Networks. *IEEE Communications Surveys & Tutorials, [Electronic ISSN: 1553-877X],* 14(2), pp. 265-278.

Eissa , T., Razak , S. & Ngadi, M. D., 2011. Towards providing a new lightweight authentication and encryption scheme for MANET. *Wireless Netw-Springer Science,* 17(1), p. 833–842.

Elhoseny, M. & Hassanien, A. E., 2018. *Dynamic Wireless Sensor Networks.* 1 ed. Switzerland: Springer International Publishing AG. In Series of Studies in Systems, Decision and Control. [eBook ISBN 978-3-319-92807-4].

Emmanouis & Christos, 2012. Security model for emergency real-time communications in autonomous networks. *Springer- Information Systems Frontiers, A Journal of Research and Innovation,* 14(3), pp. 541–553 [online]:https://link.springer.com/article/10.1007/s10796-010-9259-8.

ESSLINGER, B., 2008. *The CrypTool Script: Cryptography, Mathematics,* s.l.: www.cryptool.org.

Estebanez , A., Diego , R. & Arturo , L., 2016. New Data Structures to Handle Speculative Parallelization at Runtime. *International Journal of Parallel Programming-Springer,* 44(3), p. 407–426.

Faghihi , S., Hossein, M. & Dakhilalian, M., 2015. Biclique cryptanalysis of MIBS-80 and PRESENT-80 block ciphers. *Security and Communication Networks,* 9(1), pp. 27-33.

Ferguson, N., Kelsey, J., Lucks, S. & Schneier, B., 2001. Improved cryptanalysis of Rijndael. In: E. B . Schneier, ed. *Fast Software Encryption.* New York, USA: Springer-Verlag, pp. 213-231.

Ferrag , M. A., Maglaras , L. & Ahmim, A., 2017. Privacy-Preserving Schemes for Ad Hoc Social Networks: A Survey. *IEEE Communications Surveys & Tutorials,* 9(4), pp. 3015 - 3045.

Fluhrer, S. & Mantin, I., 2001. *Weaknesses in the Key Scheduling Algorithm of RC4.* Berlin, Springer, pp. 1-24.

Frasheri, N. & Çiço, B., 2012. *Scalability of Gravity Inversion with OpenMP and MPI in Parallel Processing.* Ohrid, Macedonia, International Conference on ICT Innovations.

Gartner, 2017. *Estimated Global Cybersecurity Spending,* s.l.: Website: https://humairahmed.com/blog/?p=8343.

GLOBALSIGN, 2018. *What is SSL,* s.l.: Globle Sign [online on]: https://www.globalsign.com/en/ssl-information-center/what-is-ssl/.

G, M. & S, M. H., 2016. *Constructing Key Dependent Dynamic S-Box for AES Block Cipher System.* India, IEEE, 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT).

Goodrich, M. & Tamassia, R., 2011. *Introduction to computer security.* 1st ed. US: pearson. [ISBN-13: 978-0321702012].

Hamada, F. & Rahman, 2016. *Impact of IPSec on MANET.* China, Computer, Consumer and Control (IS3C), 2016 International Symposium IEEE..

Hamalainen, P., Alho, T. & Hannikainen, M., 2006. *Design and Implementation of Low-area and Low-power AES Hardware core.* Croatia, Digital System Design: Architectures, Methods and Tools, 2006. DSD 2006. 9th EUROMICRO Conference. IEEE..

Hammi, B., Khatoun, R. & Zeadaly, S., 2017. IoT technologies for smart cities. *IET networks,* 7(1), pp. 1-13.

Hazzaa, F. I., Yousef, S., Sanchez, E. & Cirstea, M., 2018. *Lightweight and Low-Energy Encryption Scheme for Voice over Wireless Devices.* Washington DC, USA, IECON18 - 44th Annual Conference of the IEEE Industrial Electronics Society.

Hazzaa, F. & Yousef, S., 2017. *Performance Analysis for Traffics in Mobile Ad Hoc Network.* London, Springer International Publishing ICGS3.

Hazzaa, F., Yousef, S., Ali, N. & Sanchez, E., 2019. *Effect of Nodes Density on Real Time Traffic in MANET.* London, IEEE International Conference ICGS3.

Hercigonja, Z. & gimnazija, D., 2016. Comparative Analysis of Cryptographic Algorithms. *International Journal of DIGITAL TECHNOLOGY & ECONOMY,* 2(1), pp. 127 - 134.

Hu, D., Wang, F. & Huang, C., 2009. *A Novel Relay Encryption Scheme for Mobile Ad hoc Networks.* s.l., Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, IEEE Computer Society.

Hu, w. & Cao, March 2017. Quality-Aware Traffic Offloading in Wireless Networks. *IEEE TRANSACTIONS ON MOBILE COMPUTING,* 16(11).

Hu, Y. C., Perrig, A. & Johnson, D. B., 2003. *Rushing Attacks and Defense in Wireless Ad Hoc Networks.* s.l., s.n., pp. 30-40.

Jahankhani, H., Al-Nemrat, A. & Hosseinian, A., 2014. Cybercrime classification and characteristics. In: *Cyber Crime and Cyber Terrorism Investigator's Handbook.* s.l.:Syngress, ISBN:978-0-12-800743-3, pp. 149-164.

Jahankhani, H. et al., 2017. *The Security Challenges of the Connected World.* London, UK, Springer, 11th International Conference Global Security, Safety and Sustainability ICGS3.

Jaime , I., Usman , J., Anh, D. & Jahankhani , H., 2019. *Ransomware Impact to SCADA Systems and its Scope to Critical Infrastructure.* London, United Kingdom, IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3).

Jiehong , W. & Detchenkov, I., 2016. *A Study on the Power Consumption of Using Cryptography Algorithms in Mobile Devices.* China, Software Engineering and Service Science (ICSESS), 2016 7th IEEE International Conference .

Jonge, E. & Loo, M., 2013. *An introduction to data cleaning with R.* Hague: Statistics Netherlands.

Kak, A., 2018. *AES: The Advanced Encryption Standard,* s.l.: Lecture Notes on "Computer and Network Security".

Kepner, J. et al., 2015. *Parallel Vectorized Algebraic AES in MATLAB for Rapid Prototyping of Encrypted Sensor Processing Algorithms and Database Analytics.* Lexington, MA, USA, IEEE High Performance Extreme Computing Conference (HPEC). [sponsored by]: Assistant Secretary of Defense for Research and Engineering/ MIT Lincoln Laboratory.

Khan, A., Sun, Q. T., Mahmood, Z. & Ghafoor, A., 2017. Energy Efficient Partial Permutation Encryption on Network Coded MANETs. *Journal of Electrical and Computer Engineering,* pp. 1-10.

Khan, M., Mebrahtu, H., Shirvani, H. & Butt, J., 2017. *Manufacturing Optimization Based on Agile Manufacturing and Big Data.* University of Greenwich, UK, Proceedings of the 15th International Conference on Manufacturing Research, Incorporating the 32nd National Conference on Manufacturing Research.

KIZHVATOV, I., 2011. *Physical Security of Cryptographic Algorithm Implementations.* Luxembourg: Ph.D Thesis, University of Luxembourg.

Kolahi, S. S., Mudaliar, K. & Gu, Z., July,2017. *Impact of IPSec security on VoIP in different environments.* Milan,Italy, Ubiquitous and Future Networks (ICUFN), 2017 Ninth International Conference IEEE..

Lambić , D. & Živković, M., 2013. COMPARISON OF RANDOM S-BOX GENERATION METHODS. *PUBLICATIONS DE L'INSTITUT MATHÉMATIQUE, Ministry of Science and Technology of Serbia,* 93(107), p. 109–115.

Lee, C., 2014. Biclique cryptanalysis of PRESENT-80 and PRESENT-128. *The Journal of Supercomputing,* 70(1), pp. 95-103 doi:10.1007/s11227-014-1103-3. ISSN 0920-8542..

Liang & Chao, H.-C., 2011. Multimedia Traffic Security Architecture for the Internet of Things. *IEEE,* Issue 0890-8044/11/.

Liang & Han, 2011. Multimedia Traffic Security Architecture for the Internet of Things. *IEEE Network,* pp. 35-40.

Li, L., Keting , J. & Wang, X., 2015. Improved Single-Key Attacks on 9-Round AES-192/256. *Lecture Notes in Computer Science, book series, Fast Software Encryption,* Volume 8540, pp. 127-146.

Manchanda, N. & Anand, K., 2010. *Non-Uniform Memory Access (NUMA),* NY. USA: New York University.

Manual, F., 2015. *Guidelines for Collecting and Reporting Data on Research and Experimental Development.* Paris: The Measurement of Scientific, Technological and Innovation Activities, OECD Publishing.

Marina, B., 2009. *INTRODUCTION TO DIGITAL AUDIO CODING AND STANDARDS.* s.l.:SPRINGER NATURE (SIE).

Market.Research, 2018. *Cyber Security Market Size, Share & Trends Analysis Report,* s.l.: GRAND VIEW RESEARCH [online source]: https://www.grandviewresearch.com/industry-analysis/cyber-security-market.

Marshall, D., 2011. *Parallel Programming with Microsoft Visual.* s.l.:O'Reilly Media, Inc.

Masoud, M. ., Jannoud, I. & Ahmad, A., Sep 2015. *The power consumption cost of data encryption in smartphones.* Jordan, IEEE International Conference on Open Source Software Computing (OSSCOM).

Masoumi , M. & Rezayati, M. H., February 2015. Novel Approach to Protect Advanced Encryption Standard Algorithm Implementation Against Differential Electromagnetic and Power Analysis. *IEEE*

TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, 10(2), pp. 256-264. [available online]: (www.ieeexplore.ieee.org), Digital Library.

Massa, N., 2000. Fiber Optic Telecommunication. In: *Fundamentals of photonic.* Springfield Technical Community College: University of Connecticut/ Springfield Technical Community College, p. Modul 1.8.

Matesanz, J. G., Orozco, A. L. & Villalba, L. J., Nov. 2012. Security Issues in Mobile Ad Hoc Networks. *International Journal of Distributed Sensor Networks,* 2012(1), pp. 1-6..

Matin , M. A. & Islam, M. M., Sep.2012. *Overview of Wireless Sensor Network, Wireless Sensor Networks,* s.l.: IntechOpen, DOI: 10.5772/49376. Available from: https://www.intechopen.com/books/wireless-sensor-networks-technology-and-protocols/overview-of-wireless-sensor-network.

Matsui, M., 1994. *The First Experimental Cryptanalysis of the Data Encryption Standard.* s.l., Advances in Cryptology — CRYPTO '94. CRYPTO 1994. Lecture Notes in Computer Science, vol 839. Springer, Berlin, Heidelberg.

Memon, J., Rozan, M., Uddin, M. & Abubakar, A., 2014. Randomized Text Encryption: a New Dimension in Cryptography. *International Review on Computers and Software (I.RE.CO.S.),* 9(2), pp. 365-373.

Merkow, M. S. & Breithaupt, J., 2014. *Information Security: Principles and Practices.* 2nd ed. USA: Pearson IT Certification.

Mohammed, F. & Rohiem, A., 2009. *A novel S-box of AES Algorithm using Variable Mapping Technique.* Cairo, Aerospace Sciences & Aviation Technology.

Morgan, S., 2017. *Cyber Security Cost,* US: Forbes, available online on: https://www.forbes.com/sites/stevemorgan.

Msolli, A., Helali , A. & Maaref, H., July 2016. *Image encryption with the AES algorithm in wireless sensor network.* Tunisia, IEEE - 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP).

Muresan, R., 2012. Power Analysis Attacks and Hardware Level Countermeasures for Cryptographic Devices. *Recent Patents on Electrical & Electronic Engineering, Bentham Science Publishers,* Volume 5, pp. 173-184.

Nadeem, A. & Javed, M., 2005. *A Performance Comparison of Data Encryption Algorithms.* Pakistan, International Conference on Information and Communication Technologies IEEE.

Nagaraj, S., Raju, G. S. & Srinadth, 2015. Data Encryption and Authetication Using Public Key Approach. *Procedia Computer Sience - Elsevier,* Volume 48, pp. 126-132.

Nagendra, M. & Sekhar, M. C., 2014. Performance Improvement of Advanced Encryption Algorithm using Parallel Computing. *International Journal of Software Engineering and Its Applications,* 8(2), pp. 287-296. [availble online]: https://pdfs.semanticscholar.org/d223/ac8c99338ef031003eb2e7ac6533511bf9e5.pdf.

Navalgund, S. S., Desai, A., Ankalgi, K. & Yamanur, H., December 2013. Parallelization of AES Algorithm Using OpenMP. *Engineering and Technology Publishing, Lecture Notes on Information Theory,* 1(4), pp. 144-147.

Neuman, W. L., 2011. *Social research methods : qualitative and quantitative approaches.* 7th ed., international ed. USA: Pearson Education; National Library of Australia [available online]: https://trove.nla.gov.au/version..

NIST, 2004. *Advanced Encryption Standard,* s.l.: National Institute of Standards and Technology.

Northcutt, S., 2018. *Security Laboratory: Methods of Attack Series*, s.l.: SANS Technology Institute.

Oxford, 2017. *Letter Frequencies in English,* England: Oxford College.

P. S. Abhijith, Srivastava, M. & Mishra, A., 2013. *High performance hardware implementation of AES using minimal resources.* India, International Conference on Intelligent Systems and Signal Processing (ISSP).

Panaousis, E. A., 2012. *Security for mobile ad-hoc networks.* s.l.:British Library - Thesis (Ph.D) , Kingston University London.

Panwara, N., Sharmaa, S. & Singh, A. K., 2016. A survey on 5G: The next generation of mobile communication. *Physical Communication-Elsevier. [Since Direct.online]: www.sciencedirect.com/science/article,* 12(2), pp. 64-84.

Popov, A., 2015. *Prohibiting RC4 Cipher,* WA, USA: Internet Engineering Task Force.

Prakash, A., Satish, M., Sai, T. & G., M., Aug. 2015. Improving Cloud Security Using Multi Level Encryption and Authentication. *International Journal of Innovative Research in Information Security (IJIRIS),* 2(8), pp. 1-8.

Prasithsangaree, P. & Krishnamurthy, P., 2003. Analysis of Energy Consumption of RC4 and AES Algorithms in Wireless LANs. *GLOBECOM,* Volume 3, pp. 1445-1449.

Premkumar, P. & Shanthi, D., 2014. An Efficient Dynamic Data Violation Checking Technique For Data Integrity Assurance In Cloud Computing. *International Journal of Innovative Research in Science, Engineering and Technology,* 3(3).

Preneel, B. & Rijmen, V., 2000. *Comments by NESSIE project on the AES finalists,* s.l.: National Institute for Standard and Technology (NIST), [available online]: http//:csrc.nist.gov.

purevpn, 2017. *Here's How You Can Secure Your Wireless Network,* s.l.: https://www.purevpn.com/blog/securing-wireless-network-guide/.

Rahma, A. S. & Yaco , B. Z., 2012. Real-Time Partial Encryption of Digital Video using Symmetric Dynamic Dual Keys Algorithm. *Engineering and Technology Journal,* 30(5), pp. 710-728.

RAMESH, G. & UMARANI, R., 2012. Performance Analysis of Most Common Symmetrical Encryption Algorithms. *International Journal of Power Control Signal and Computation(IJPCSC),* 13(1), pp. 42-45.

Rana, B. & Wankhade, S., 2017. Hybrid Cryptographic Algorithm for Enhancing Security of Text. *International Conference On Emanations in Modern Technology and Engineering,* 5(3), pp. 339-443.

Riad, A., Elminir, H., Shehata, A. & Ibrahim, T., 2013. SECURITY EVALUATION AND ENCRYPTION EFFICIENCY ANALYSIS OF RC4 STREAM CIPHER FOR CONVERGED NETWORK APPLICATIONS. *Journal of ELECTRICAL ENGINEERING,* 64(3), pp. 196-200.

Rouse, M., 2014. *application security,* s.l.: TechTarget. [available]: https://searchsoftwarequality.techtarget.com/definition/application-security.

Rukhin, A., Soto, J., Nechvatal, J. & Samid, M., April 2010. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications,* USA: National Institute of Standards and Technology/ U.S. Department of Commerce.

Sahu, S. K. & Kushwaha, A., 2014. Performance analysis of Symmetric Encryption algorithm for Mobile ad hoc networks. *International Journal of Emerging Technology and Advanced Engineering,* 4(6), pp. 619-624.

Salama, D. & Hadhoud, M., 2010. Evaluating the effect of symmetric Algorithms on Power consumption for Different Data types. *International Journal of Network Security,* 11(2), pp. 78-87.

Salomon, D., 2012. *Data Compression.* 2 nd ed. New York: Springer.

Samad , S. K., Mudaliar, K. & Zhang, C., July 2017. *Impact of IPSec security on VoIP in different environments.* Milan, Italy, IEEE-Ninth International Conference on Ubiquitous and Future Networks (ICUFN).

Sandoval & Garc´ıa, 2012. Security Issues inMobile Ad Hoc Networks. *International Journal of Distributed Sensor Networks,* pp. 1-6.

Savas , E. & Koc, C., 2010. Finite field arithmetic for cryptography. *IEEE Circuits and Systems Magazine,* 10(2), pp. 40-56.

Scarfone, K., Souppaya , M. & Sexton, M., Novmber 2007. *Guide to Storage Encryption Technologies for End User Devices,* US: National Institute of Standards and Technology/ Information Technology Laboratory.

Scarfone, K., Souppaya, M. & Sexton, M., 2007. *Guide to Storage Encryption Technologies for End User Devices,* US: National Institute of Standards and Technology/ Information Technology Laboratory.

Schneier, B., 2015. *Applied Cryptography: Protocols, Algorithms and Source Code in C.* s.l.:Wiley; 20th Anniversary edition, ISBN-13: 978-1119096726.

Sehgal, A., Ahuja , R. & Kumari , S., 2011. *Security architecture for Mobile Ad Hoc Networks.* s.l., Proc. of the International Conference on Science and Engineering (ICSE 2011).

Sharma , P., Liu, H., Wang, H. & Zhang, S., 2017. *Securing wireless communications of connected vehicles with artificial intelligence.* Waltham, MA, USA, IEEE International Symposium on Technologies for Homeland Security (HST).

Singh, P., Fabruary 2012. AES Keys and Round Functions for Data Security. *International Journal of Computer Applications (0975 – 8887),* 39(11), pp. 23-27.

Smith, R., 2003. *Understanding encryption and cryptography basics*, s.l.: Information Security magazine [online]: (https://searchsecurity.techtarget.com) .

SSL, 2015. *EAVESDROPPING ATTACK: A DARK SHADOW ON THE NETWORK,* s.l.: Website: SSL Shop [online access]: https://www.cheapsslshop.com/blog/eavesdropping-attack-a-dark-shadow-on-the-network.

Stallings, W., 2012. *Cryptography and Network Security Principles and Practice.* 5th ed. England: Pearson.

Stallings, W., 2017. *Cryptography and Network Security: Principles and Practice (7th Edition).* 7 th ed. Harlow- England: Pearson Education Limited.

Suriya, d. & Rajamani, 2015. Detecting and Preventing Sybil Attacks in Wireless Sensor Networks Using Message Authentication and Passing Method. *The Scientific World Journal (Hindawi).*

Techbast, K., 2016. *IP Security IPsec*, s.l.: TECHBAST, [online: https://techbast.com/].

Thomas , G. & Robertazzi, 2017. *Introduction to Computer Networking.* 1 ed. Switzerland: Springer International Publishing AG.

Thomas, . L., Pantelis , A. & Loukas, . G., 2010. *Wireless Network Traffic and Quality of Service Support: Trends and Standards (Premier Reference Source) Hardcover – 30 Jun 2010.* 1st ed. USA: Engineering Science Reference [ISBN-13: 978-1615207718].

Trapnell, B. & French, C., 2016. *Cryptographic Module Validation Program,* USA: National Institute of Standards and Technology NEST.

Trichina, E., Kerkishko, T. & Lee, H., 2005. *Advanced Encryption Standard - AES.* 1 ed. Germany: Springer-Verlag Berlin Heidelberg.

UK.gov, 2018. *UK defence and security export statistics for 2017*, UK: Gov.UK/ Depatment of International Trade.

Umaparvathi, 2010. *Evaluation of symmetric encryption algorithms for MANETs.* India, IEEE International Conference on Computational Intelligence and Computing Research.

Usman, M., Yang, N. & Jan, M., 2018. A Joint Framework for QoS and QoE for Video Transmission over Wireless Multimedia Sensor Networks. *IEEE Transactions on Mobile Computing,* 17(4).

Veena, Valiaveetil, D. R. & Gopinath, N. K., 2016. A Lightweight High-Performance Fault Detection Scheme for the Advanced Encryption Standard and Scalable Encryption Algorithm using Composite Fields. *International Journal of Engineering Trends and Technology (IJETT),* 37(7), pp. 357-361. ISSN: 2231-5381/ [online]: http://www.ijettjournal.org .

Waggoner, B., 2010. *Compression for Great Video and Audio.* 2nd edition ed. Newyork and London: Routledge.

Wang, J., Gao, Q., Cheng, P. & Yu, Y., 2014. Lightweight Robust Device-Free Localization in Wireless Networks. *IEEE Transactions on Industrial Electronics, IES,* 61(10).

Watkinson, J., 2001. *Art of Digital Audio.* Third Edition ed. U.S.A: Focal Press.

Website, 2018. *Data Encryption Standard,* s.l.: Tutorials Point website: https://www.tutorialspoint.com/cryptography/data_encryption_standard.htm.

Wikipedia, n.d. *Man-in-the-middle attack,* s.l.: Wikipedia.

Ye , G. & Huang, X., 2016. An Image Encryption Algorithm Based on Autoblocking and Electrocardiography. *IEEE MultiMedia,* 23(2), pp. 64-71.

Yick, J., Mukherjee, B. & Ghosal , D., 2008. Wireless sensor network survey. *Computer Networks [Contents lists available at ScienceDirect]- Elsevier,* Volume 52, p. 2292–2330.

Zhang, P. & Lin, C., 2014. A Lightweight Encryption Scheme for Network-Coded Mobile Ad Hoc Networks. *IEEE Transactions on Parallel and Distributed Systems,* 25(9), pp. 2211 - 2221.

Zhang, P., Lin, C. & Yixin, 2014. A Lightweight Encryption Scheme for Network-Coded Mobile Ad Hoc Networks. *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS,* 25(9), pp. 2211-2220.

Zhao, H. V., Lin, W. S. & Liu, K., Jan. 2009. A Case Study in Multimedia Fingerprinting: Behaviour Modelling and Forensics for Multimedia Social Networks. *IEEE Sig. Proc. Mag,* 26(1), p. 118–39.

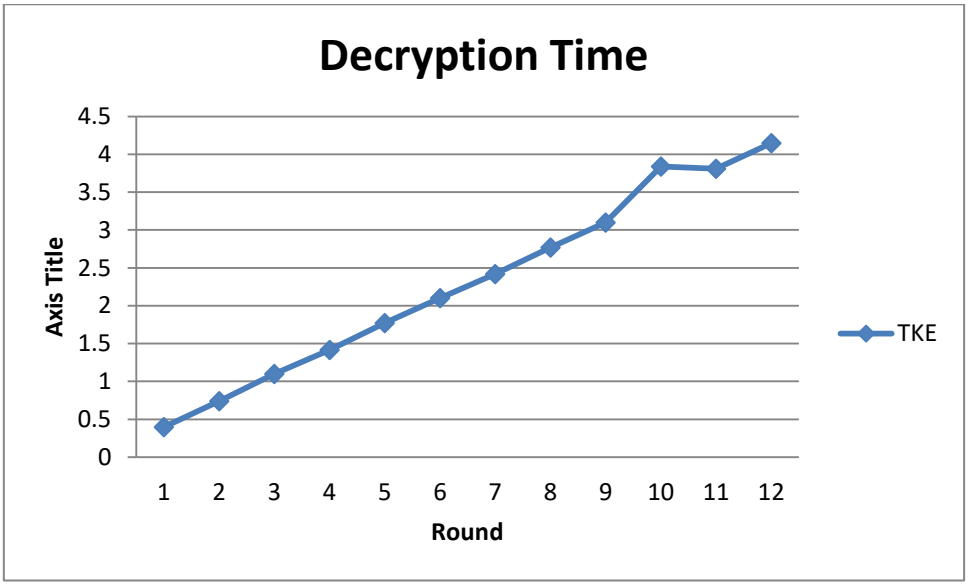
Zheng, Z., Cai, L. X. & Shen, X., 2013. *Sustainable Wireless Networks.* 1 ed. Switzerland: Springer International Publishing, ISBN:978-3-319-02468-4.

Zhou, L., 2010. Distributed Scheduling Scheme for Video Streaming over Multi-Channel Multi-Radio Multi-Hop Wireless Networks. *IEEE JSAC,* Volume 28, p. pp. 409–19.

Appendix A

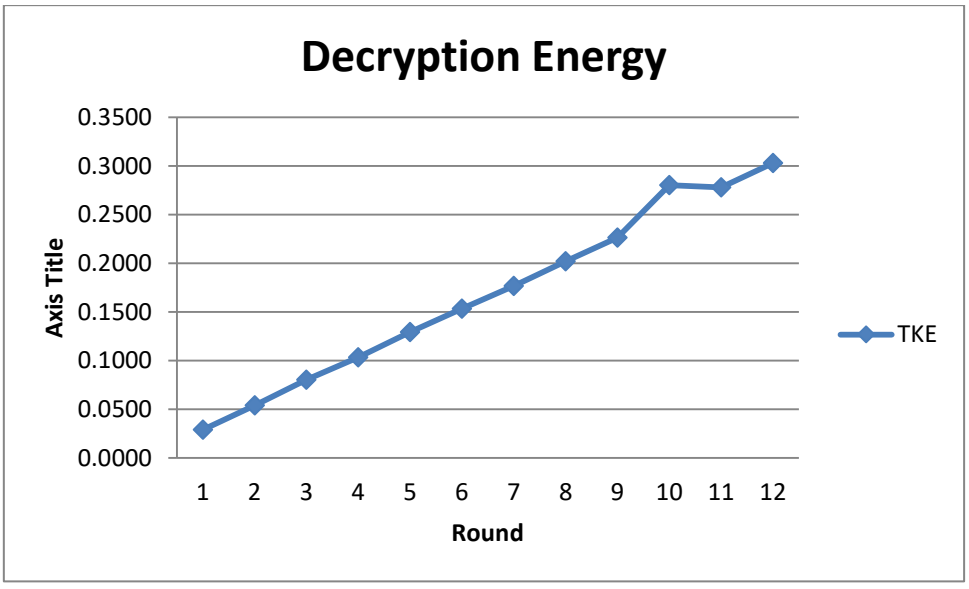
Cryptography Performance Time of (computer file) for Many Rounds TKE

Rounds Iteration	Encryption Time (Sec)	Decryption Energy (µ J)
1st Rn	0.393	0.398
2nd Rn	0.736	0.74
3rd Rn	1.07	1.1
4th	1.412	1.417
5th	1.765	1.77
6th	2.087	2.1
7th	2.417	2.42
8th	2.774	2.77
9th	3.109	3.1
10th	3.487	3.48
11th	3.804	3.81
12th	4.148	4.15



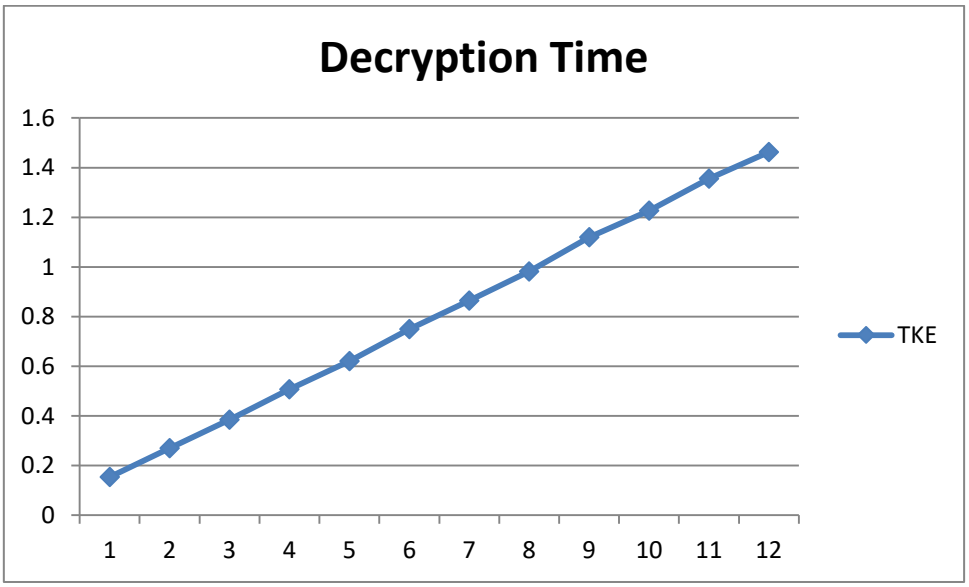
Cryptography Performance Energy of (computer file) for Many Rounds TKE

Rounds Iteration	Encryption Energy (µ J)	Decryption Energy (µ J)
1st Rn	0.0287	0.0291
2nd Rn	0.0537	0.0540
3rd Rn	0.0781	0.0803
4th	0.1031	0.1034
5th	0.1288	0.1292
6th	0.1524	0.1533
7th	0.1764	0.1767
8th	0.2025	0.2022
9th	0.2270	0.2263
10th	0.2546	0.2803
11th	0.2777	0.2781
12th	0.3028	0.3030



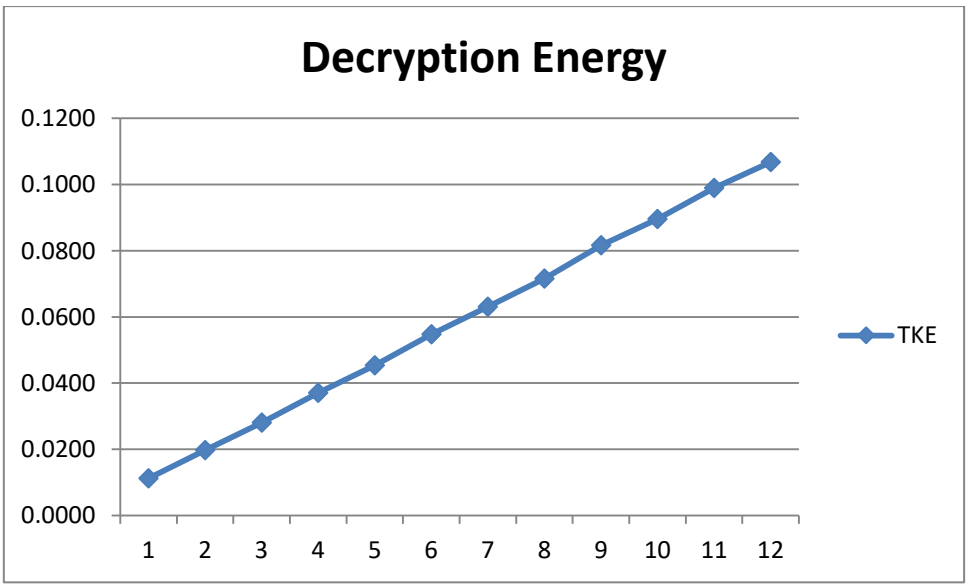
Decryption Time of (teaching file) for Many Rounds TKE

Rounds Iteration	Decryption Time (Sec)
1st Rn	0.154
2nd Rn	0.27
3rd Rn	0.384
4th	0.507
5th	0.621
6th	0.75
7th	0.864
8th	0.981
9th	1.119
10th	1.227
11th	1.355
12th	1.463



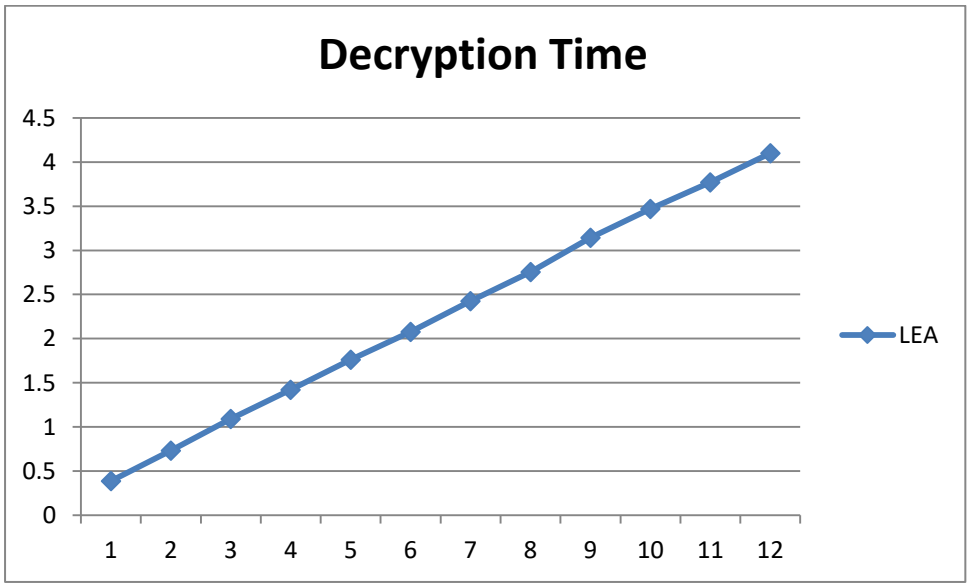
Decryption Energy of (teaching file) for Many Rounds TKE

Rounds Iteration	Decryption Energy (µ J)
1st Rn	0.0112
2nd Rn	0.0197
3rd Rn	0.0280
4th	0.0370
5th	0.0453
6th	0.0548
7th	0.0631
8th	0.0716
9th	0.0817
10th	0.0896
11th	0.0989
12th	0.1068



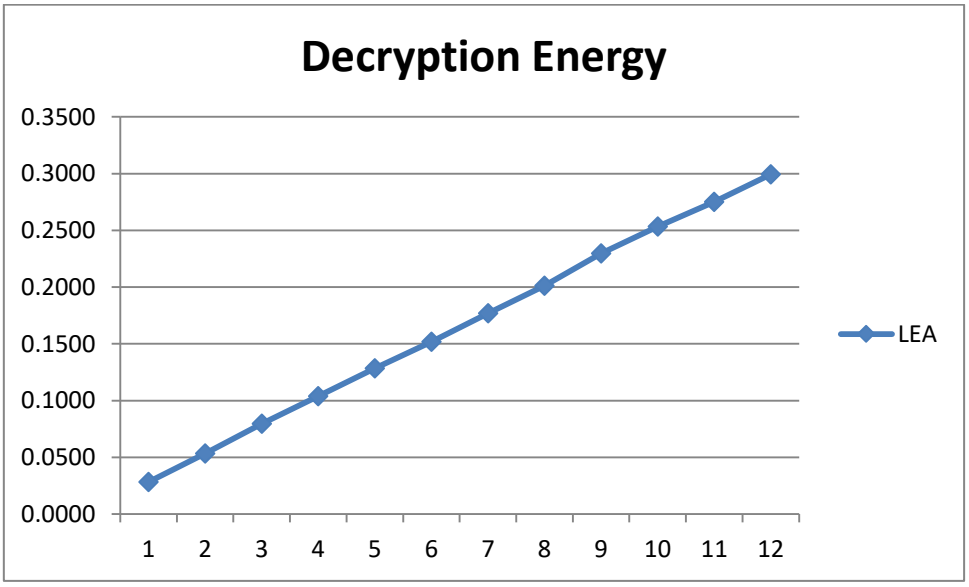
Decryption Time of (computer file) for Many Rounds LEA

Rounds Iteration	Decryption Time (Sec)
1st Rn	0.387
2nd Rn	0.73
3rd Rn	1.09
4th	1.421
5th	1.76
6th	2.078
7th	2.424
8th	2.757
9th	3.145
10th	3.47
11th	3.77
12th	4.1



Decryption Energy of (computer file) for Many Rounds LEA

Rounds Iteration	Decryption Energy (µ J)
1st Rn	0.0283
2nd Rn	0.0533
3rd Rn	0.0796
4th	0.1037
5th	0.1285
6th	0.1517
7th	0.1770
8th	0.2013
9th	0.2296
10th	0.2533
11th	0.2752
12th	0.2993



Appendix B: Network Performance

Investigating the QoS metrics for MANET connected to the Internet

The aim of this experiment is to find out the delay and throughput of a different kind of traffic in MANET with and without connection to the Internet.

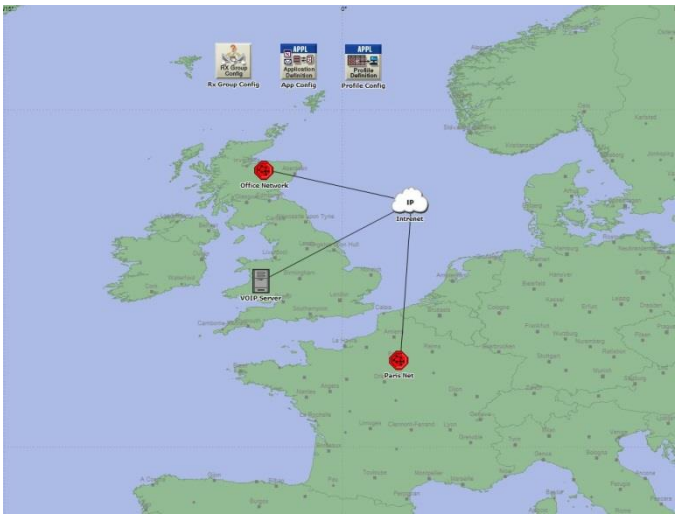
Simulation setup and parameters

The OPNET 18.5 modeler and the wireless suite are used in our simulation. 50 nodes are involved for the creation of topology on 1 km x 1 km space. Different traffic is chosen: FTP, Voice, and Video Conference. The effect of different traffic is tested.

The table below shows the simulation parameters of the network. 10 simulations have carried out and the duration 60 min for each.

Simulations Parameters

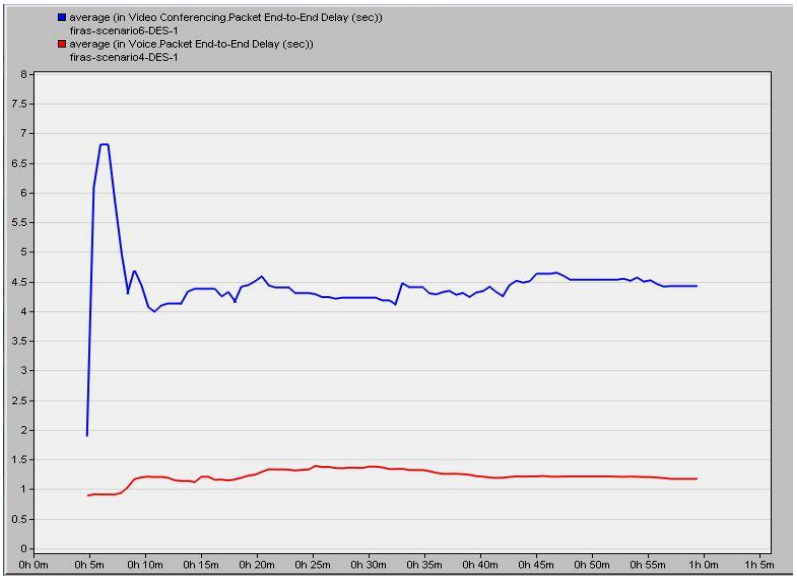
No. of simulations	10	**Simulation Area**	1000*1000 sq.m.
Simulation Duration	60 min each	**Data Rate**	11 Mbps
No. of Nodes	50	**Transmission Power**	0.001 w
WLAN Physical Characteristic	802.11	**Buffer Size**	32 KB
Freq. Band	2.4	**Routing Protocol**	AODV
Packet Size	512	**Traffic**	FTP, Voice, Video Conference
Fragmentation Threshold	1024	**Simulation**	50000 events
Simulation Seeds No.	182	**Values per statistic**	100



Network Topology

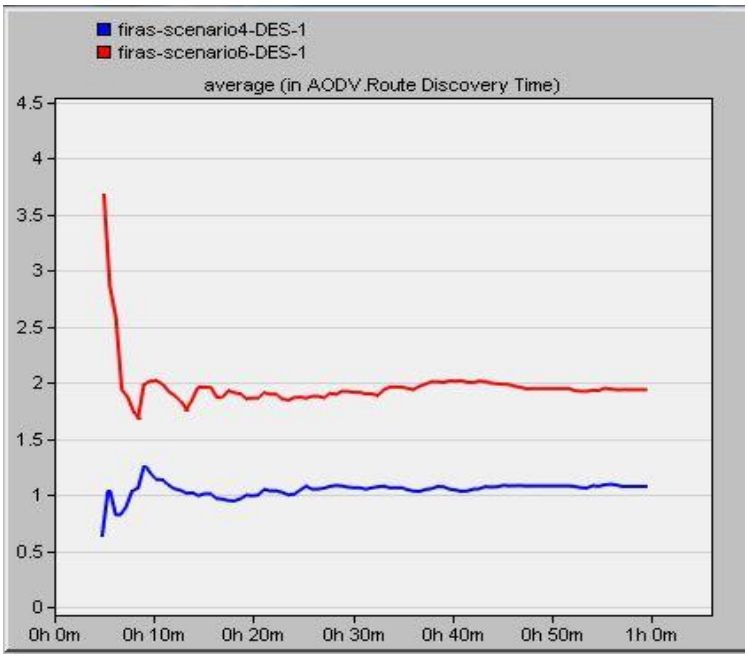
Simulation Results and analysis

The following figures show the simulation results for the QoS parameters: delay and jitter in the network.



(a) End to End delay (sec)

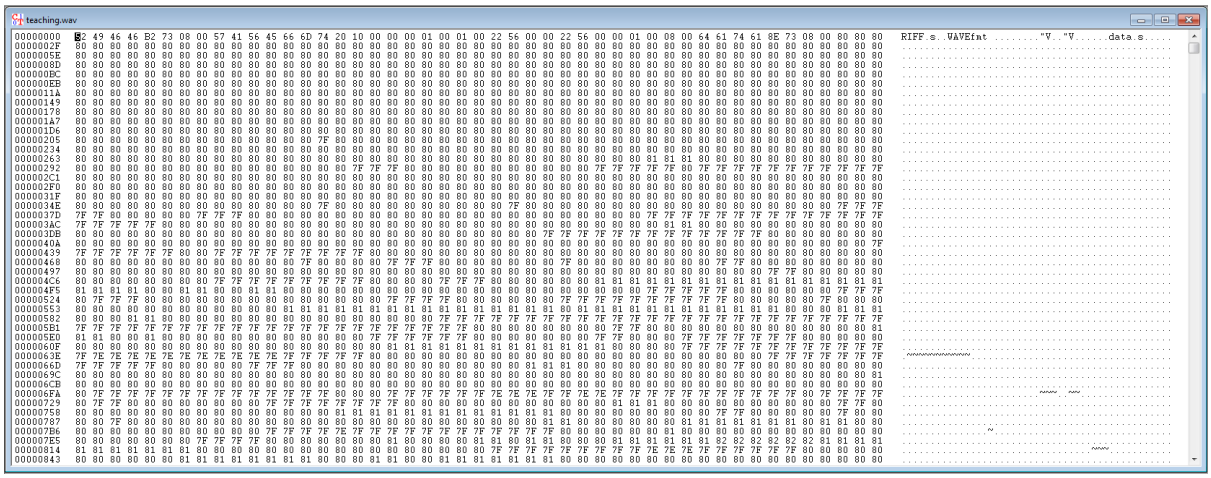
(b) voice jitter



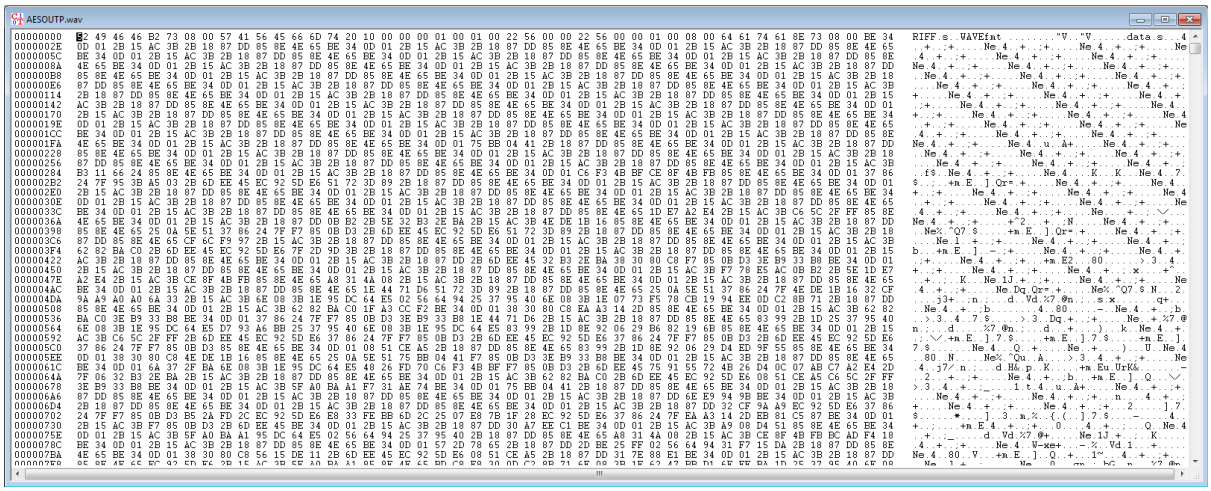
(b) Route Discovery Time

Appendix C ： CrypTool Cryptography Analysis Results

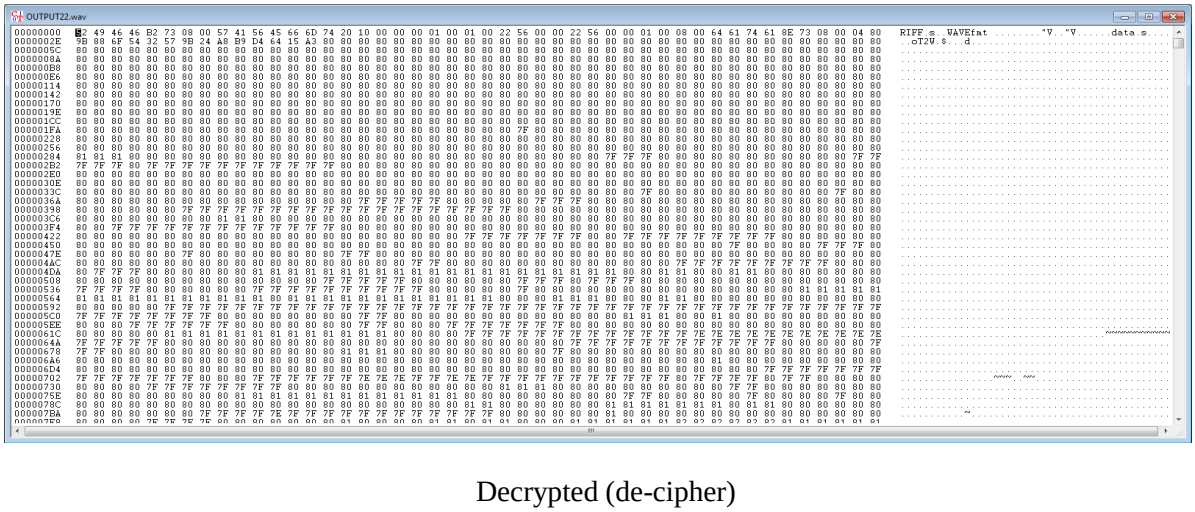
AES implementation on teaching.wav file:



Plain file

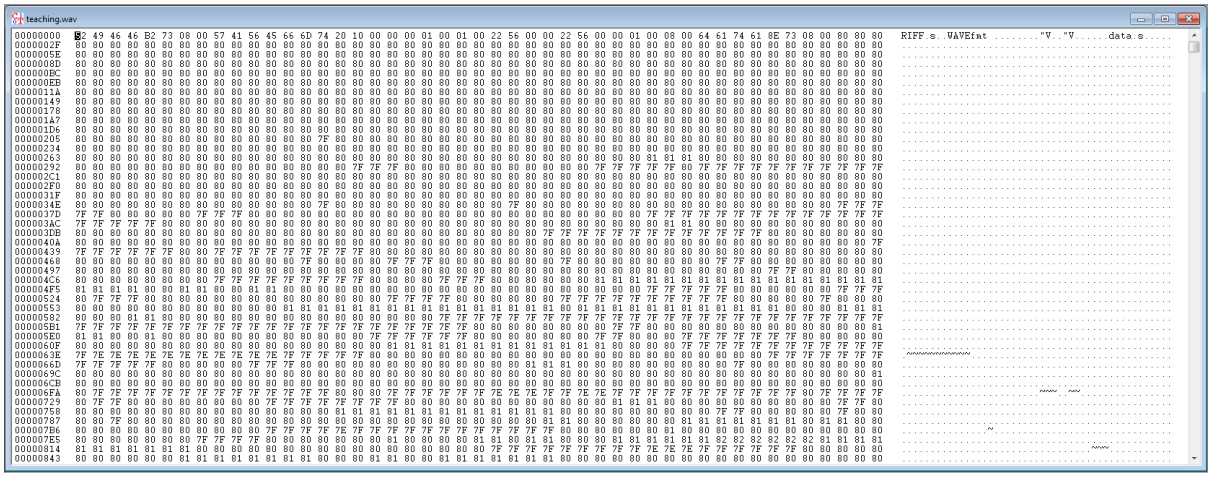


Encrypted (Cipher)

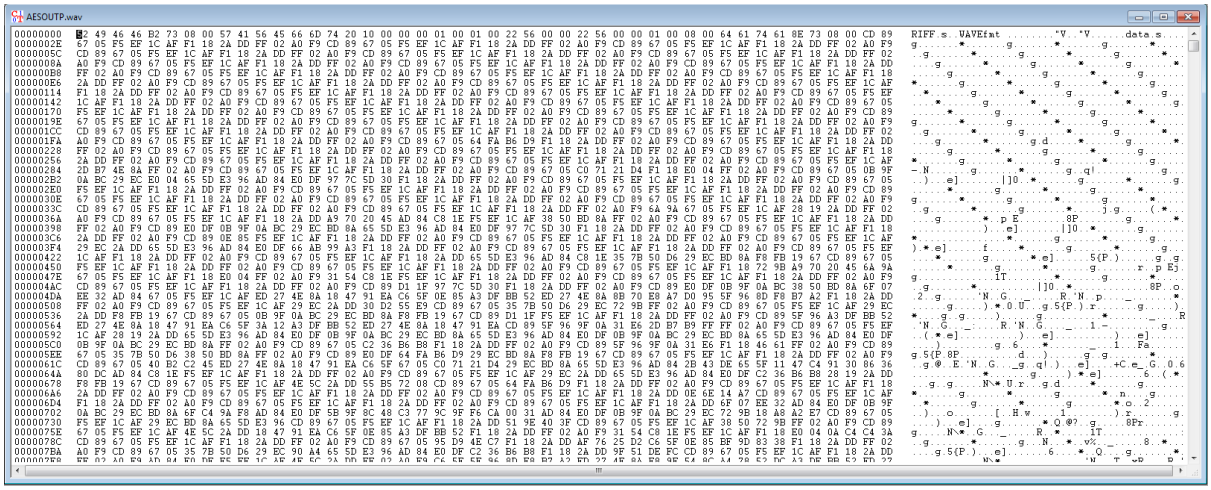


Decrypted (de-cipher)

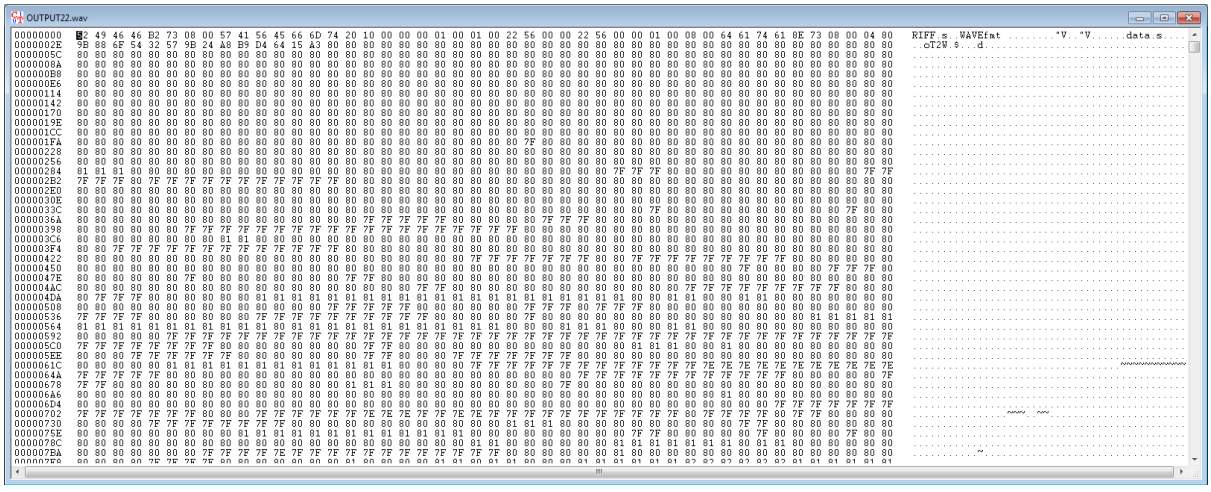
TKE implementation on teaching.wav file:



Plain file

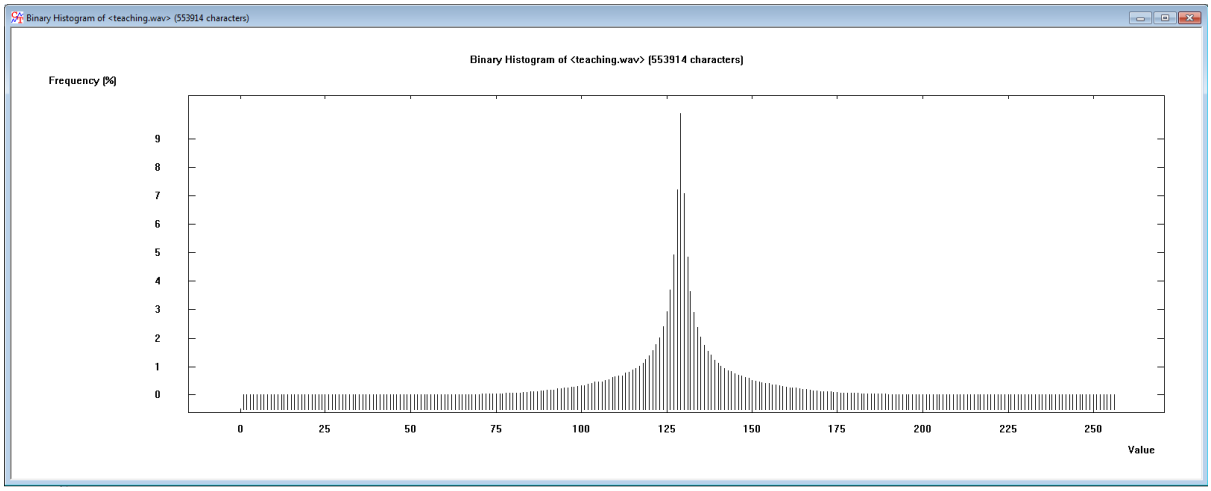


Encrypted (Cipher)

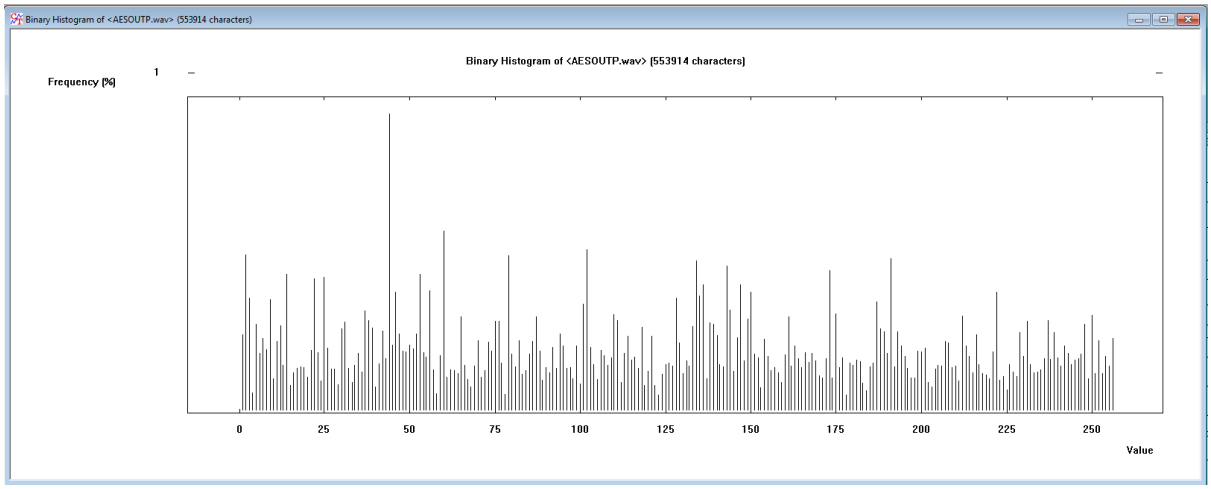


Decrypted (de-cipher)

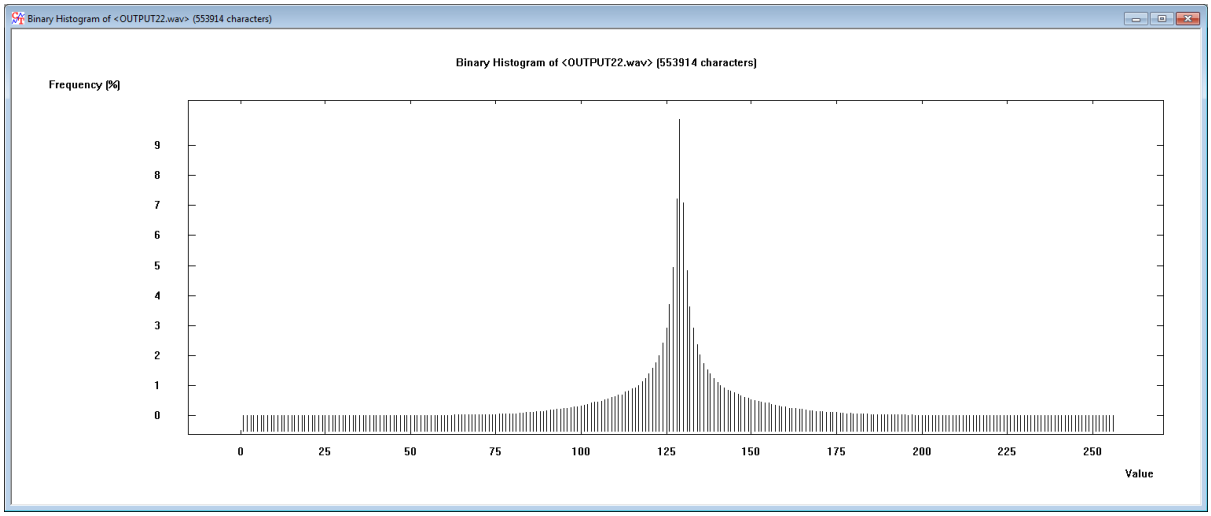
Binary Histogram for AES encryption:



Plain file

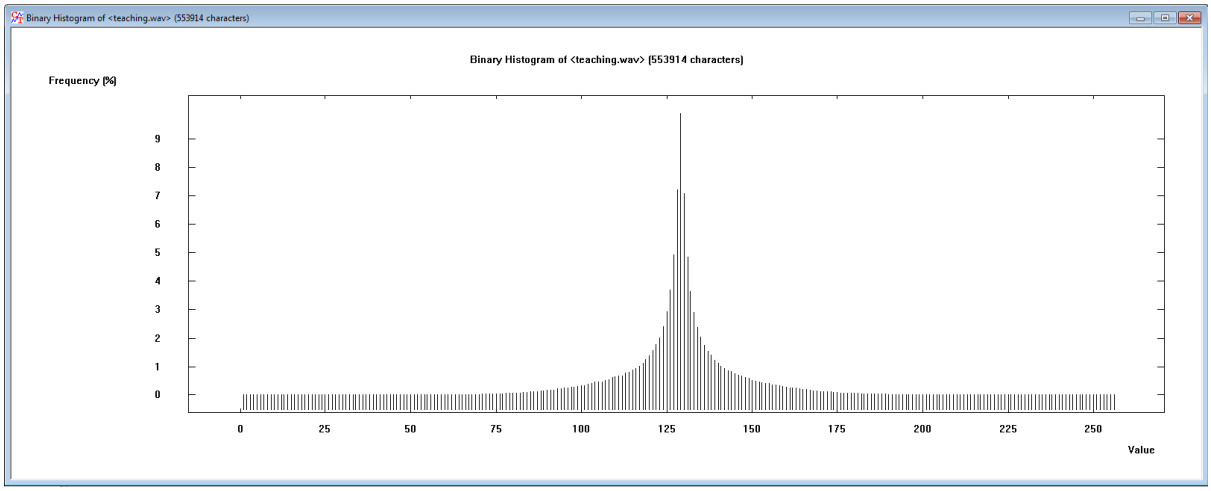


Encrypted (Cipher)

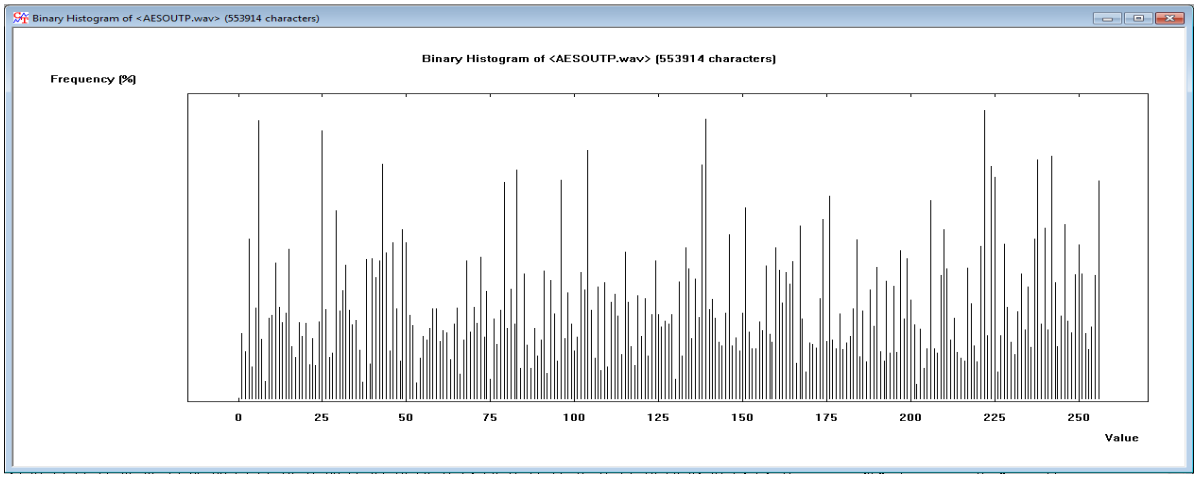


Decrypted (de-cipher)

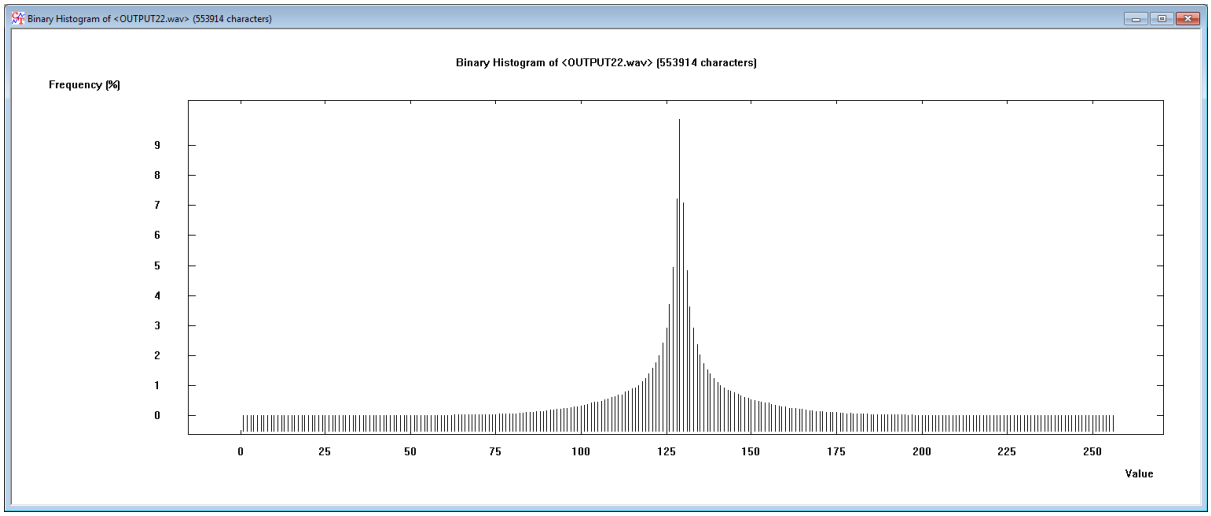
Binary Histogram for TKE encryption:



Plain file

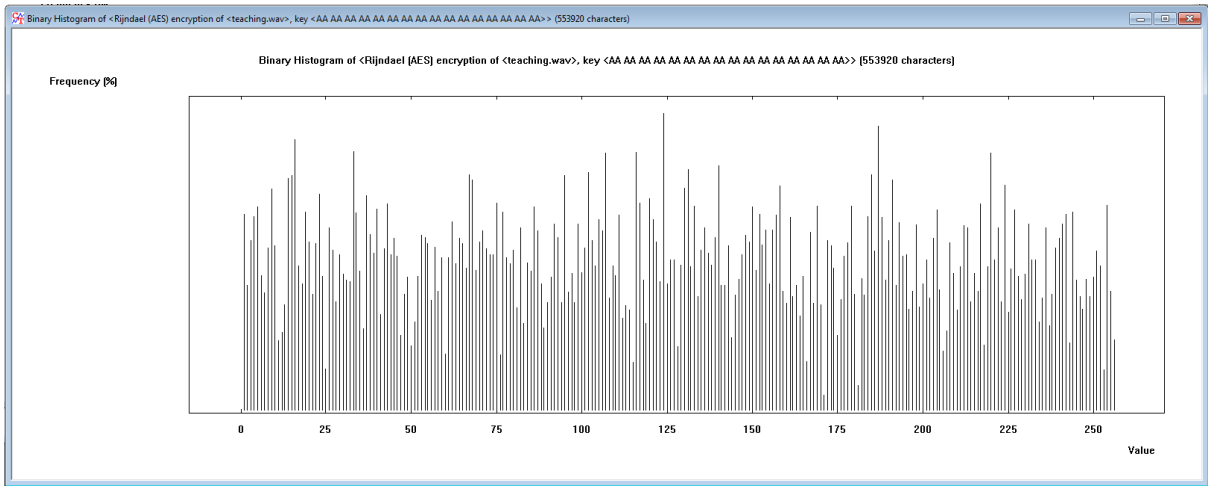


Encrypted (Cipher)

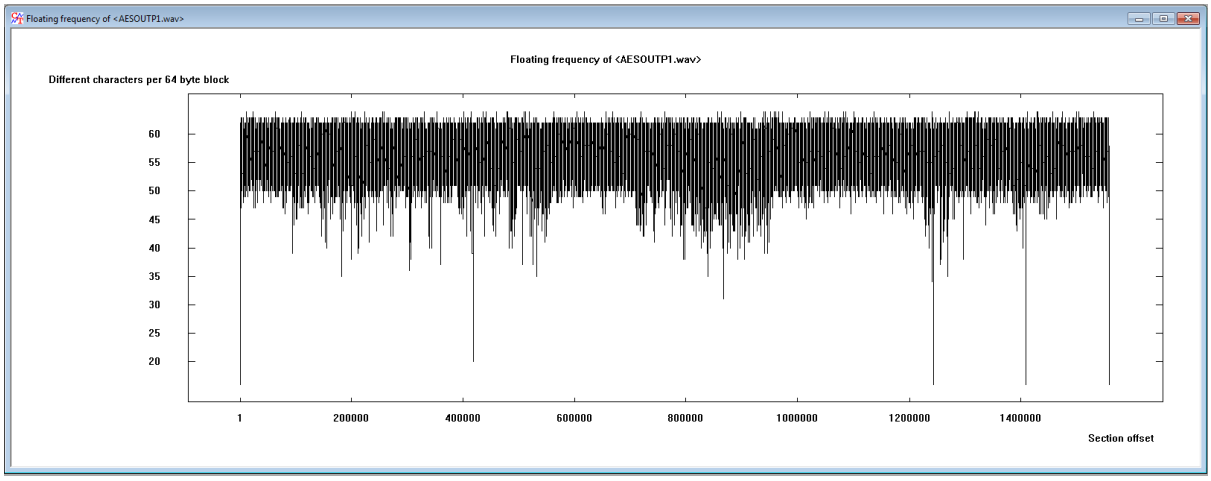


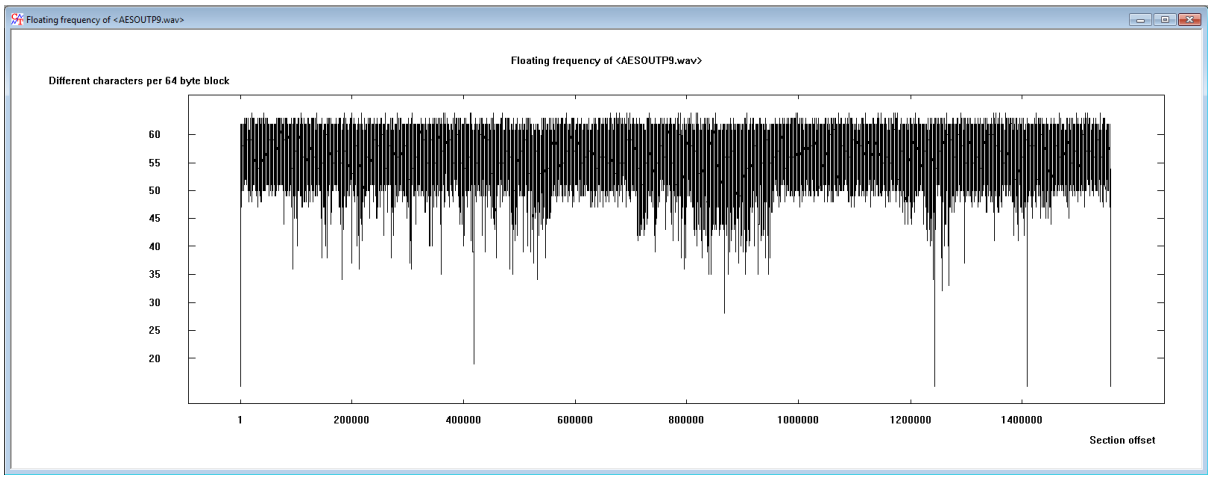
Decrypted (de-cipher)

AES encryption using cryptool

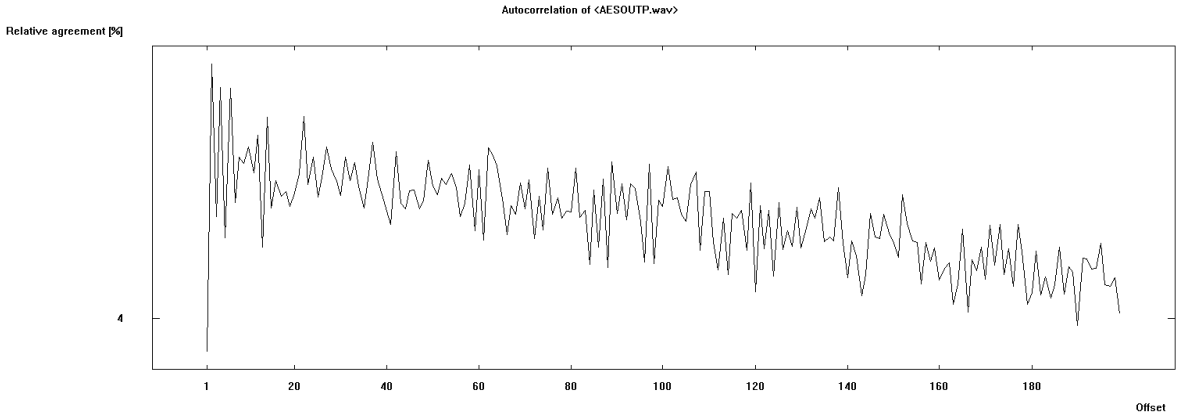


Floating frequency

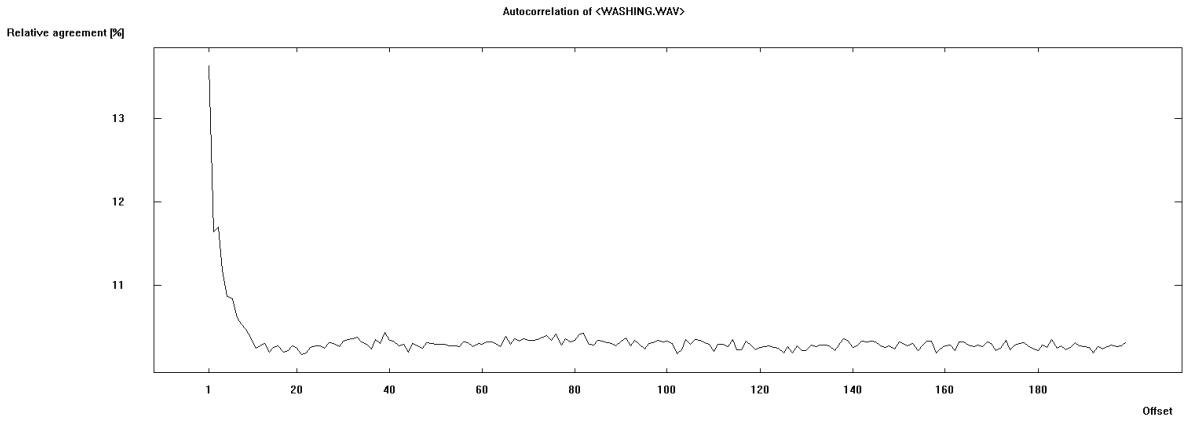




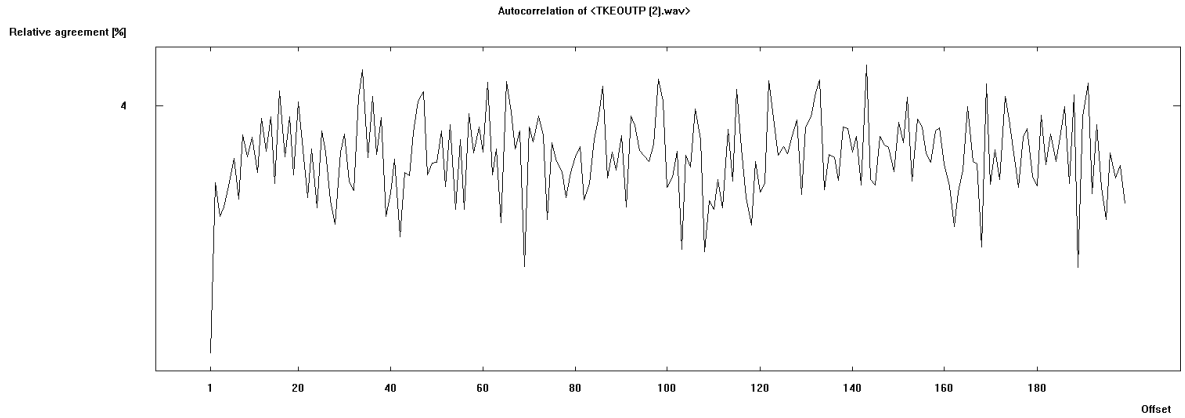
Autocorrelation:



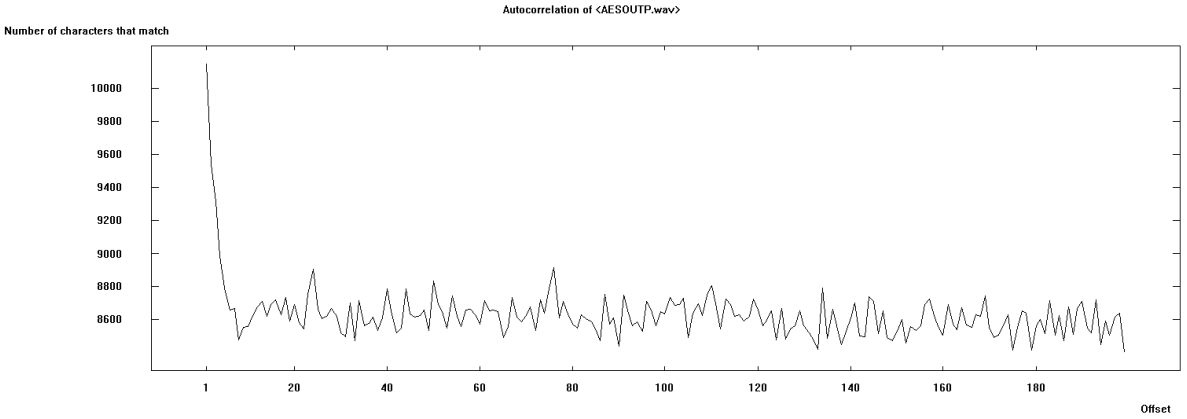
Autocorrelation Relative agreements for Cipher file (test.wav) by AES



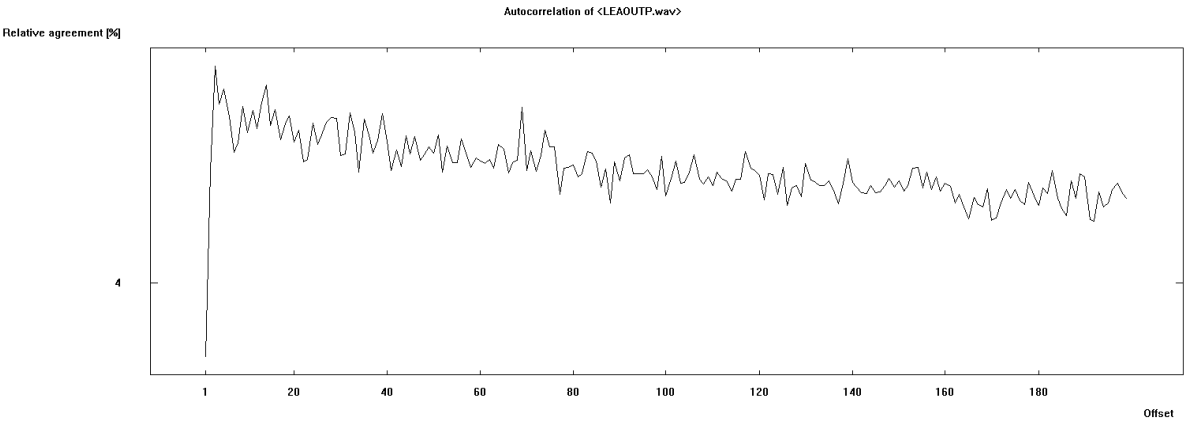
Autocorrelation Relative agreements for Plain file (washing.wav)



Autocorrelation Relative agreements for Cipher file (washing.wav) by TKE



Autocorrelation for Cipher file (washing.wav) by AES



Autocorrelation Relative agreements for Cipher file (washing.wav) by LEA

List of Publication

- F. Hazzaa, S. Yousef, E. Sanchez and M. Cirstea, "Lightweight and Low-Energy Encryption Scheme for Voice over Wireless Devices," *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, Washington, DC, 2018, pp.2992-2997.doi:10.1109/IECON.2018.8591451 https://ieeexplore.ieee.org/document/8591451

- F. Hazzaa, S. Yousef, N. H. Ali and E. Sanchez, "The Effect of Nodes Density on Real Time Traffic in Mobile Ad Hoc Network," *2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3)*, London, United Kingdom, 2019, pp. 209-212. https://ieeexplore.ieee.org/document/8688314

- F. Hazzaa, S. Yousef, " Performance Analysis for Traffics in Mobile Ad Hoc Network," *Springer International Publishing 11th International Conference on Global Security, Safety and Sustainability (ICGS3)*, London, United Kingdom, 2017

- F. Hazzaa, "Real Time Traffic in Mobile Ad Hoc Network" (*Poster*) *6th Annual Research Conference*, Anglia Ruskin University, Chelmsford, UK, 2016.

- Hazzaa, 2017, Poster, 7th Annual Research Conference, Anglia Ruskin University, Chelmsford, UK.

- Firas Hazzaa, S. Yousef, (*Accept*) Book chapter: "*New Issues on Cyber Security Forensics*", Cyber Security Practitioner's Guide, WORLD SCIENTIFIC PUBLISHING COMPANY PTE LTD, Singapore, Feb 2019

- Firas Hazzaa, S. Yousef, N. Ali, A. Shabut, (*ReView*) "*A Novel Security and Cryptography Scheme for Real Time Traffic over Wireless Networks*", Journal of Information Security and Applications, 2019, Ref: JISA_2019_122

Research Funding Observatory (RFO) Seminar

The Research Funding Observatory is a series of focused, targeted events and seminars to support academic and research staff at our University on the path to success.

The session, entitled 'Funder focus: UK Research Councils', will provide an overview of new developments as a result of creating the new body 'UK Research and Innovation', with a focus on how Research Councils support funding in the UK and their main funding schemes.

This seminar will also cover the different types of funding calls on offer and the key information and documents required when developing a Research Council proposal.

Please only book on the seminar if you're a member of academic or research staff at our University as these events are not open to the general public.

The seminars will take place on both campuses at the following times:

Chelmsford
Wednesday 7 November 2018, 12:15 – 13:15
SAW 105

Cambridge
Thursday 8 November 2018, 12:15 – 13:15
YST 110

Please book your attendance via Eventbrite.

For any further information contact: observatory@anglia.ac.uk

PhD Researcher presents at conference

Firas Hazzaa, from our School of Engineering & the Built Environment, presented his research in Cyber Security, at the 44th Annual Conference for IEEE Industrial Electronic Society IECON18, in Washington D.C, USA, during October.

The research paper had been accepted for publication in the conference proceedings and Firas was invited by the conference committee to attend and present it, along with co-authors Prof Marcian Cirstea, Head of our School of Computing & Information Science, and Firas' supervisors, Dr Sufian Yousef and Dr Erika Sanchez-Velazquez.

Firas explains, "the aim of our research is to address the energy consumption gaps during the encryption processes for the real time traffic over wireless networks."

During the event Firas managed to meet many experts in his field from all over the world who also presented their innovative technology and research.

For more details about the conference visit www.iecon2018.org

For more information contact: firas.hazzaa@pgr.anglia.ac.uk

Left to right: Firas Hazzaa and co-author, Prof Marcian Cirstea

****

Firas Ibrahim Hazzaa, MSc, MIEEE, ChPP
Ph.D. Research Postgraduate
Faculty of Science and Engineering
Anglia Ruskin University
Bishop Hall Lane
Chelmsford
Essex
CM1 1SQ
UK
+44 (0)7424 511 690
firas.hazzaa@pgr.anglia.ac.uk